

INTERIM REPORT

---

**UNDERSTANDING FINANCIAL  
REPORTS USING NATURAL  
LANGUAGE PROCESSING**

---

January 20, 2019

**Supervisor:** Dr. RuiBang Luo

Tarun Sudhams  
Varun Vamsi Saripalli

## **Abstract**

Credit Derivatives are considered excellent tools to hedge the credit risk of an underlying entity from one party to another without actually transferring the ownership of the entity. One such hedging tool is called Credit Default Swaps (CDS), which are often known to be responsible for the 2007-2008 financial crisis. Upon further investigation, it was found that lack of regulation and information on how CDS works were the main culprits behind the crisis. Post the crisis, the United States Securities and Exchange Commission (SEC) has requested for frequent and more detailed reporting from the mutual funds about their current position on these derivatives. Given the lack of strict format for the report, it becomes extremely difficult to extract information from these reports and conduct in-depth analysis on how the mutual funds leverage credit derivatives and in particular, CDS.

This project aims at consolidating all the mutual fund holding reports on Credit Default Swap positions and investigate how mutual funds leverage credit derivatives by first scrapping all the publicly available reports from the SEC and structure the data for Natural Language Processing to understand the context of the sentences used in the reports and then conduct more downstream analysis (Time Series Analysis) to extract key insights from the structured data.

Furthermore, this report aims at providing a detailed motivation behind our project and also in explaining some of the key technologies that enable us to successfully create a database to empower future research studies and also go over the key challenges and limitations faced by our team. Finally, we would look at the current progress and goals achieved since the inception of this project.

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Background . . . . .	5
1.2	Objective . . . . .	6
1.3	Scope . . . . .	6
1.4	Deliverables . . . . .	7
1.5	Outline of the report . . . . .	7
<b>2</b>	<b>Literature Review</b>	<b>8</b>
2.1	Use of CDS by Corporate Bond Funds . . . . .	9
2.2	Mutual Fund Holding of CDS . . . . .	9
2.3	Bidirectional LSTM - CRF for Adverse Drug Event Tagging of Electronic Health Records . . . . .	9
2.4	Time Series Analysis . . . . .	11
<b>3</b>	<b>Methodology</b>	<b>13</b>
3.1	Data Preprocessing . . . . .	14
3.1.1	Data Collection . . . . .	14
3.1.2	Building Corpus . . . . .	14
3.1.3	Parsing . . . . .	21
3.1.4	Preparing Training Data . . . . .	22
3.2	Natural Language Processing . . . . .	24
<b>4</b>	<b>Current Status and Results</b>	<b>26</b>
<b>5</b>	<b>Difficulties Encountered</b>	<b>27</b>
<b>6</b>	<b>Future Plan</b>	<b>28</b>
<b>7</b>	<b>Conclusion</b>	<b>28</b>
	<b>References</b>	<b>30</b>

## **List of Algorithms**

1. Folder Restructure Pseudocode
2. Data Categorization
3. Corpus Generation

## **List of Figures**

1. Deep Learning Architecture used in the study
2. Flow of information in a LSTM Module
3. Proposed Workflow
4. Updated File Structure
5. Directory of Structured Folder
6. Corpus File Structure
7. Example of an extracted N-Q report
8. Defining Labels in the Tool
9. Tagging in Action
10. An example of tagged sentence in CoNLL 2003 format with NER(Named Entity Recognition) tags
11. Mathematical representation of BiLSTM Layer
12. Conditional probability of output sequence
13. F1 Score on Train and Test Data

# **1 Introduction**

The lack of a structured database of financial reports makes it difficult for Credit Default Swap related research studies to conduct a much more comprehensive and quantitative analysis and also result in inaccurate case studies when it comes to critical topics like predicting the next financial crisis. Therefore, a structured and well maintained database can help future research papers to analyze CDS and retrieve new and exciting information from it.

## **1.1 Background**

This project investigates how mutual funds leverage credit derivatives by studying their routine filings to the U.S. Securities and Exchange Commission. Credit Derivatives have a wide range of products and we will be studying a class of credit derivatives called Credit Default Swaps(CDS). Credit Default Swaps have a reference entity linked to them which are generally governments or corporations. The buyer has a credit asset with the reference entity and buys a CDS from the seller to insure himself against a default in the payment by the reference entity. It is thus used as a hedging tool to reduce the risk associated with a credit asset [6]. The buyer makes periodic payments to the seller till the date of the maturity of the contract and this constitutes the spread of the CDS. In the event of a credit default, the seller has to pay the buyer of the CDS the face value of the credit asset and all the interest payments that the buyer would have earned between that time till the date of the maturity of the asset.

Credit default swaps are traded over the counter and hence there isnt much information available on it. The forms filed by the Mutual Funds regarding their CDS activities were in an unorganized manner before SEC had requested for more frequent and detailed fund holdings at the end of 2016. This resulted in it being extremely difficult to get relevant information from these reports to

carry out further analysis. Thus, such information regarding CDS is extremely valuable as it would provide transparency and can be used to set appropriate capital requirements. There have been a few previous studies exploring the usage of CDS by Mutual Funds[1],[13] but these reports examined only a small number of the institutions over a short period of time. Hence, we choose to comprehensively examine all the reports from 2004- 2016 and this makes the results of our project extremely valuable for further research.

## **1.2 Objective**

The objective of this project is to successfully run Natural Language Processing algorithms and analyze, get credible insights from the reports extracted from the US Securities and Exchange Commission Website. This analysis on the data about the Credit Default Swaps usage by Mutual Funds will be contained in a consolidated database. We will then build an interactive website which will give future researchers easy access to this database and help in future research projects.

## **1.3 Scope**

The project is divided into two parts — crawling the reports from the US Securities and Exchange Commission website through the years from 2003-2017 and then analyzing the data from the reports. The crawling of the reports will result in around 150 GB of data that will have to be processed to get the relevant information. The scope of this project was designed such that it is extremely comprehensive in nature. Hence we decided to scrape all the publicly available SEC EDGAR website and use it for our analysis. Unlike the similar study conducted earlier (Wei and Zhu, 2016), which only took into account of reports filed between 2007 — 2011. This project aims at analyzing all the data publicly available which is from 2004 — 2016 and use it to conduct a significant downstream analysis and draw insights from it.

## **1.4 Deliverables**

Through this project, we aim to get data of Mutual Funds holdings of CDS from 2003-2017 and to perform comprehensive analysis on it. The first part of this project is to identify the proportion of mutual funds using CDS. Next, we will identify the proportion of filings that are easy to extract ,the proportion of filings that need to be extracted using NLP and the remaining proportion of filings that cannot be effectively extracted. Furthermore, we have to extract the characteristics of the CDS holdings which will include the identifier of the mutual fund(CIK,name), metadata. The particulars of the CDS holdings are also extracted which will contain the reference entity, direction of trade, notional amount, currency, contract premium and the counterparty. Downstream analysis will then have to be performed on the extracted CDS Information to explore patterns of CDS usage and to possibly predict the usage during certain economic conditions

## **1.5 Outline of the report**

This report would briefly go through the motivation and background behind the project and would also further delve into the methodologies to cater to both technical and non-technical audience. Furthermore, this report also aims to be report the current status of the project and the future milestones that are planned to achieved. Finally, the report will conclude with an in-depth look on the challenges and limitations that we have faced and how the team plans to tackle it.

## **2 Literature Review**

While we conducted our research to find studies which used similar techniques of Natural Language Processing to extract key semantics in the data, we found two studies in particular which were very similar to the niche topic that we are targeting as well, namely Financial Industry.

The first study goes over the pragmatic research approaches on computational linguistics that allows them to tackle the problem of financial forecasting (Xing, Cambria Welch, 2017). They are using similar techniques of web scraping (also called text mining) and structuring the data collected to conduct predictive analysis and forecast future events. However, a key issue with the study is that the data collected during the collection process includes a lot of noise or inaccuracies which in turn leads to unreliable predictions. One of the key goals of our study is to learn from this study and avoid making the same error.

The second study also involves structuring of data retrieved from policy papers published by the governments and public bodies across the globe. Just like the problem we face the study identifies that more than 80% of the data retrieved from these papers is unstructured making it extremely difficult to conduct any sort of analysis (Ferati, 2017). The research question for this study includes exploring the techniques that could be used to make this structured database for future analysis of policies. However, the study mostly looks into theoretical details of the question with very little practical tests conducted to refute the conclusions and results. Hence this study could be used as a great starting point to learn about the techniques that our study could use.

Finally, we take a look at a study which involves extracting adverse drug events by tagging the Electronic Health Records using Bidirectional LSTM-CRF and examine its implementation and potential insights which can be carried out from it.



## 2.1 Use of CDS by Corporate Bond Funds

CDS has been used in multiple research papers to understand how multiple factors affect the maximum revenue that can be generated from investment in CDS. One such interesting paper discusses how complex strategies which involve a team of CDS team-managed funds outperforms a solo-managed team due to the abundance of diverse skill and opinions in the team. But at the same time they might perform poorly if the market conditions are extremely dynamic and quick decisions are required (Adam Gutter, 2015).

## 2.2 Mutual Fund Holding of CDS

A similar research to ours was conducted to understand the effects of CDS on pre and post financial crisis capital markets by using mutual funds holdings of CDS contracts between the period of 2007-2011 (Wei Zhu, 2016). The research concluded that reference entities which had the highest grossing interest revenue were actually the ones which deemed as too big to fail. The metadata present in the CDS reports enables research studies to come up with interesting and meaningful interpretations which in turn allow major investors in CDS to make much more informed decisions.

## 2.3 Bidirectional LSTM - CRF for Adverse Drug Event Tagging of Electronic Health Records

Sequence Tagging refers to the pattern-recognition problem which involves labelling each member of the sentence to a pre-assigned label. The labelling task itself can be treated as an independent classification problem for one member per task. Electronic Health Records are a great example of containing both structured and unstructured data which can be difficult to extract with just the use of rule-bases extraction tools like *Regular Expressions* and hence a need for a deep learning framework is required to learn from the patterns observed in the unstructured data.

To mitigate this, the study has described a three-layered deep learning architecture which consists of a single layer of character-level word representation using bidirectional LSTM(Long Short Term Memory, another layer of bidirectional LSTM for extracting the contextual meaning and final layer of CRF(Conditional Random Field) for predicting the label for each member in the sentence.

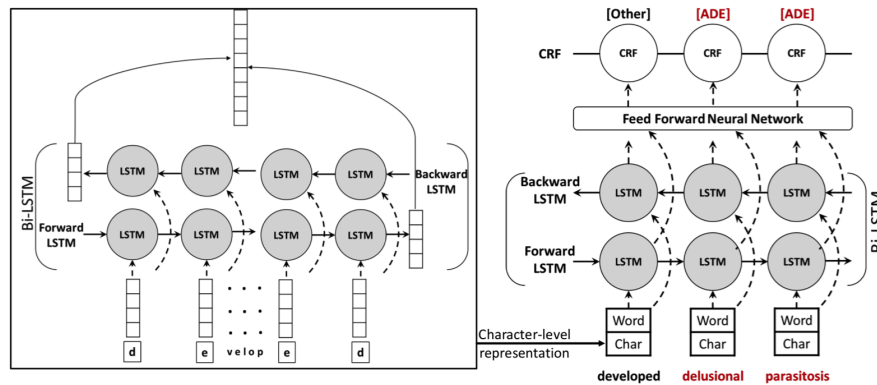


Figure 1: Deep Learning Architecture used in the study

Similar to our methodology, the study has made use of various datasets to verify the implementation of their algorithm and then use their manually tagged 1,084 notes to train the model. Unlike our use case, the study was able to find preannotated datasets and hence required little to no data pre-processing. The evaluation set used by them consisted of 213 EHR(Electronic Health Records) notes and yielded them a *F1 score* of 0.8373, 0.8454, and 0.841 of phrase level execution. This goes to show that in order to extract unstructured data from such a large corpus would require a deep learning framework to yield accurate result of extraction.

However, given the flexible and ever updating nature of our data, it is imperative for us to preprocess and weed out any data that is beyond the scope of the project which includes reporting of any asset class other than Credit Default Swap.

## 2.4 Time Series Analysis

The volume of data that is being structured in this process makes it a perfect use case of time series analysis. Time Series Analysis helps us find complex pattern in data which has time as one of its categorical features. Given the nature of data that we are analysing, we are trying to observe the data on a quarterly basis and therefore we can implement Long Short Term Memory (LSTM) on top of RNN to learn the long term dependencies of a specific categorical feature (Olah, 2015). This further enhances the memory state of the already existing LSTM and makes it great for language learning, handwriting recognition tasks which requires identifying key patterns in the reference object. After NLP, auto-encoders would allow us to derive key insights on the structured data and help us make useful predictions based on the patterns detected in the data. (Bansal, 2018)

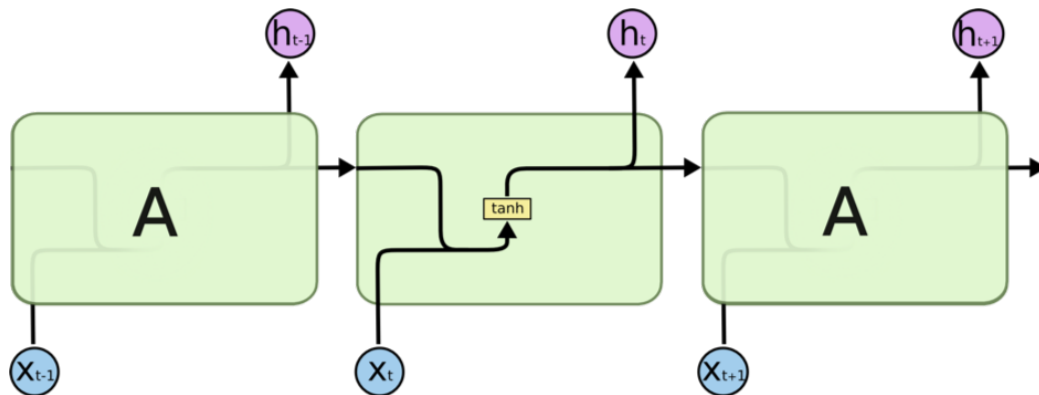


Figure 2: Flow of information in a LSTM Module

LSTM or Long Short Term Memory are a special type of Recurrent Neural Network which are capable of learning long-term dependencies. Traditionally, neural networks are not capable of remembering conclusions that they might have drawn from previous analyses (Colah, 2016). Each time you throw a problem at them, they start working on it from scratch. That's where LSTMs play a vital role. It allows neural networks to remember the conclusions from previous analyses and use it in future downstream analysis.

This makes them a great tool for our use case as their implementation would allow us to remember the meanings of the words learnt from the word vectors that we figured in our language algorithms. Figure 2 goes through a simple repeating module of LSTM that shows us how the information in the LSTM module persists from one module to another.

### 3 Methodology

The workflow for the entire project can be divided into three steps namely the data preprocessing, implementing Natural Language Algorithms and Downstream Analysis on extracted data. Figure 3 gives a good overview of our project and each step has been explained in detail in the subsequent parts below.



Figure 3: Proposed Workflow

In a nutshell, the web crawling is done using Python and Perl scripts to collect gigabytes worth of financial reports from the United States Securities and Exchange Commission website will include our first step namely data collection process. Then we will move on to our natural language processing step which will help us convert the unstructured data collected in step 1 into a structured database. This database would help us achieve one of the goals of our study. Furthermore, with the help of this database, we will conduct a time series analysis which will allow us to analyze the structured data and make useful predictions like predicting the next financial crisis. Finally, the

database also aims to be a resource for future research papers on Credit Default Swaps by helping them conduct thorough and streamlined analysis.

## **3.1 Data Preprocessing**

### **3.1.1 Data Collection**

This is the first step of our project which serves as the data collection process. Given the nature of our research subject, it is quite easy to locate the data as everything is available publicly so the need to employ key web scraping tools to consolidate the entire report filings from 2004-2016 arises. There are few mature technologies which could be employed for this purpose such as NodeJS, C/C++, Python, PHP and Perl. However, there have been stability issues with NodeJS as it makes deploying multiple crawlers a complicated job. Development costs for C/C++ is very high and PHP has a very bad scheduler scheme which makes it harder to work with (Koshy, 2017). These reasons helped us identify Python and Perl as the best programming languages to write our scripts in.

### **3.1.2 Building Corpus**

In order to build a corpus for our NLP tasks, it was important for us to restructure the file structure such that it is easy to navigate between different kinds of reports. We had around 146 GB of data in one SEC folder which contained all the Funds' N-CSR, N-CSRS and N-Q reports along with their CIK number. The initial folder structure was that of the main SEC-Edgar-Data folder containing all the folders with the Mutual Funds names which in turn had a folder named with the Funds' CIK number which then had the folders N-CSR, N-CSRS, N-Q in them which respectively contained the Funds financial reports from 2003-2017. After going through the data, it was found that few of the Mutual Funds had changed the Fund names through the time period and this resulted in there being duplicate files under different Fund names but with the

same CIK number. We first wrote a script to restructure the data such that we could split it into 3 parts from which we could extract the data sequentially.

---

**Algorithm 1:** Folder Restructure Pseudocode
 

---

**Result:** Restructured Folder

initialization;

**for** all the directories in SEC-Edgar-Data folder **do**

**for** all the directories in the Fund Name folder **do**

**if** the directory is N-CSR **then**

            Move the entire sub-directory to the N-CSR Folder created  
            outside SEC-Edgar-Data;

**end**

**if** the directory is N-CSRS **then**

            Move the entire sub-directory to the N-CSRS Folder created  
            outside SEC-Edgar-Data;

**end**

**if** the directory is N-Q **then**

            Move the entire sub-directory to the N-Q Folder created  
            outside SEC-Edgar-Data;

**end**

**end**

**end**

---

The above algorithm goes over the pseudocode of the script which was run on the original folder and it segregated the original folder into three folders namely N-CSR, N-CSRS and N-Q. Each of these folders contained a folder with the CIK number and inside that folder was the folder with the Mutual Funds name containing the respective reports. In the event that a Mutual Fund had changed its name during the time period of 2003-2017, then this restructuring made it easier for us to identify the Fund by its unique CIK number . After restructuring , the folder structure was as follows(similar

structuring for N-CSRS and N-Q):

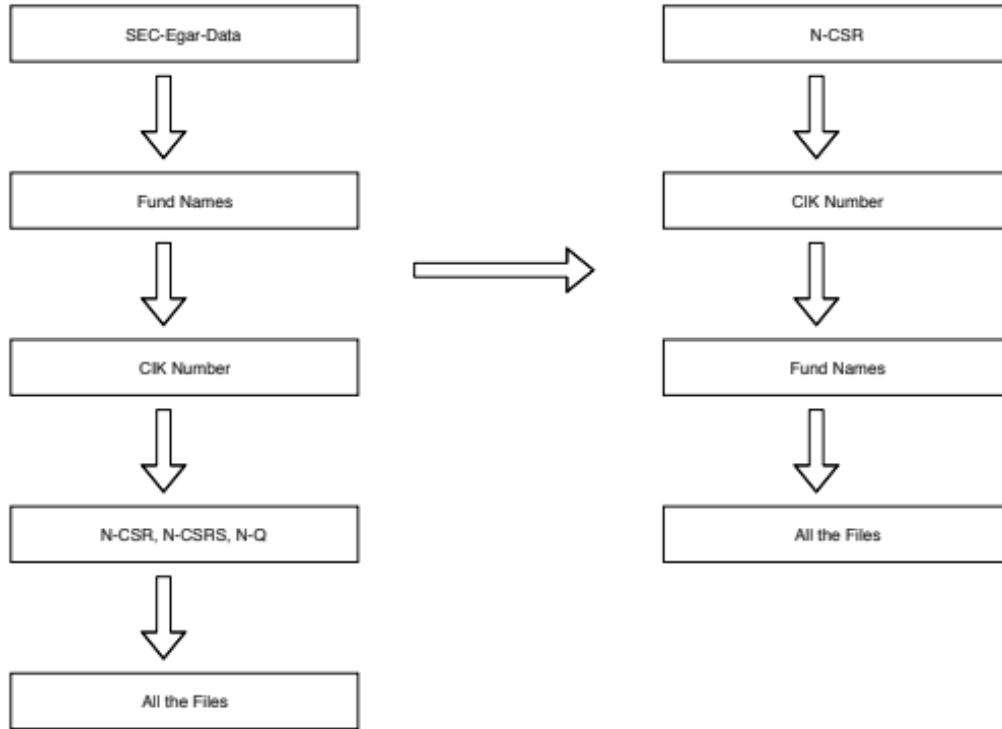


Figure 4: Updated File Structure



The second part of the restructuring process was figuring out the reports which had unstructured data and the reports which had structured data which could be extracted. We could extract the structured data by running a *Perl* script which would extract the relevant CDS information from the reports. We wrote another script which would run the Perl script on all the files that we had.

It would then redirect the output to a temporary text file. We then checked the contents of the text file to determine if there was extracted information in the temporary text file. If there was information in it, we moved it to a new folder called Structured. If it was unstructured and the temporary text file didn't have relevant information in it, we moved those reports to the unstructured folder. This way, we had a smaller set of folders on which to run our NLP model.

---

**Algorithm 2:** Data Categorization

---

**Result:** Data categorized into structured and unstructured format initialization;

```
for all the directories in N-CSR folder do
|
|   for all the directories in the CIK Number do
|   |
|   |   for each file under Fund Name do
|   |   |
|   |   |   Run the perl script on the file and redirect output to a
|   |   |   temp.txt file;
|   |   |   if there is required output in the temp.txt file then
|   |   |   |   Move the original file to Structured folder & remove
|   |   |   |   temp.txt file;
|   |   |   else
|   |   |   |   Move the original file to unstructured folder & remove
|   |   |   |   temp.txt file
|   |   |   end
|   |   end
|   end
end
end
```

---

The directory structure for the structured folder is:

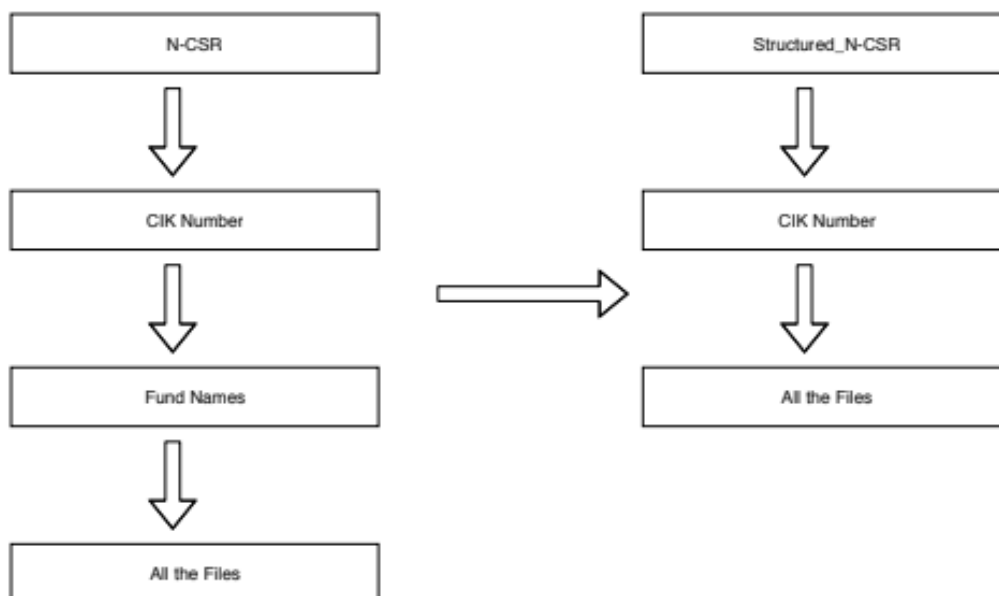


Figure 5: Directory of Structured Folder

The folder structure for the unstructured folder is similar. We then unzipped all the files in the unstructured folder to their original text files. After going through the data in the unstructured folder, we found out that along with files in there which had unstructured CDS information that couldnt be extracted with the Perl script, there was also files which didnt have any mentions of Credit Default Swap. We hence decided to run another script to remove these files and cut down the size of our dataset to only the useful reports to speed up the processing time. This script checked all the text files in the unstructured folder for mentions of the word Credit Default Swap and when it found it, moved the respective text file to a new folder called CDS which contained all the relevant files. The files with no mentions were moved to another folder called NoCDS.

**Algorithm 3:** Corpus Generation**Result:** Final Corpus Created

initialization;

**for** all the directories in unstructured N-CSR folder **do**    **for** each file under CIK Number **do**

Run the perl script on the file and redirect output to a temp.txt file;

**if** there are mentions of Credit Default Swaps or relevant information **then**

Move the original file to Corpus N-CSR &amp; rename file to include the path in the name;

**else**

Move the original file to No\_CDS folder;

**end**    **end****end**

Folder Structure for the Corpus Folder which contains all the files that we are going to use: The above scripts were applied to N-CSR, N-CSRS, N-Q.

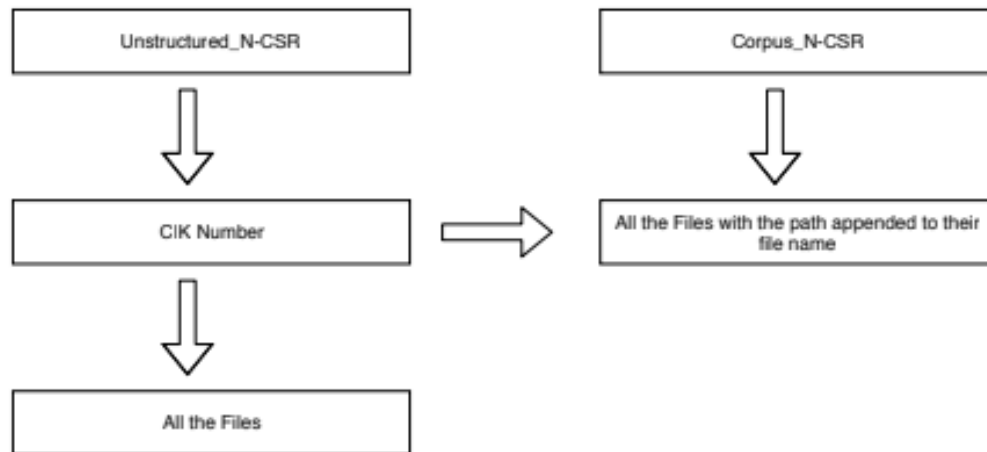


Figure 6: Corpus File Structure

We hence had 3 separate CDS folders at the end containing the useful reports of N-CSR, N-CSRS and N-Q. Once we had these folders, we had to start tagging these reports manually to be able to run our NLP model on them. To make the tagging process easier, we built a Text Annotation tool using Django with Python which would allow us to import our text files, tag the files according to the labels that we created which would be the Reference Entity, Counterparty, Notional Amount etc. We could then export the tagged data into a csv file.

### **3.1.3 Parsing**

Given the enormous corpus built through crawling the SEC's EDGAR database, it resulted in most of the files still have XML tags in them which was not of relevance to the financial data that we wanted to extract. In order to mitigate this we developed python scripts using *BeautifulSoup* and *HTMLParser* libraries to get rid of the XML tags. This helped us to reduce the file size of each text file for upto 17% which in turn resulted in reduction of meaningless data from the corpus and also relatively smaller size of it.

```

0001193125-17-056504 2.txt
<SEC-DOCUMENT>0001193125-17-056504.txt : 20170224
<SEC-HEADER>0001193125-17-056504.hdr.sgml : 20170224
<ACCEPTANCE-DATETIME>20170224170355
ACCESSION NUMBER:      0001193125-17-056504
CONFORMED SUBMISSION TYPE: N-Q
PUBLIC DOCUMENT COUNT: 2
CONFORMED PERIOD OF REPORT: 20161231
FILED AS OF DATE:      20170224
DATE AS OF CHANGE:     20170224
EFFECTIVENESS DATE:   20170224

FILER:

COMPANY DATA:
COMPANY CONFORMED NAME:      PIMCO FUNDS
CENTRAL INDEX KEY:          0000810893
IRS NUMBER:                  952632339
STATE OF INCORPORATION:     MA
FISCAL YEAR END:            0331

FILING VALUES:
FORM TYPE:                    N-Q
SEC ACT:                      1940 Act
SEC FILE NUMBER:              811-05028
FILM NUMBER:                  17638283

BUSINESS ADDRESS:
STREET 1:                     650 NEWPORT CENTER DRIVE
CITY:                          NEWPORT BEACH
STATE:                          CA
ZIP:                            92660
BUSINESS PHONE:               949-720-6000

MAIL ADDRESS:
STREET 1:                     650 NEWPORT CENTER DRIVE
CITY:                          NEWPORT BEACH
STATE:                          CA
ZIP:                            92660

<SERIES-AND-CLASSES-CONTRACTS-DATA>
<EXISTING-SERIES-AND-CLASSES-CONTRACTS>
<SERIES>
<OWNER-CIK>0000810893
<SERIES-ID>S000009675
<SERIES-NAME>PIMCO All Asset All Authority Fund
<CLASS-CONTRACT>
<CLASS-CONTRACT-ID>C000026506

```

Figure 7: Example of an extracted N-Q report

### 3.1.4 Preparing Training Data

This serves as the last process of the Data Preprocessing step which involves manually tagging hundreds of reports. As this a cumbersome process, extreme precision and proper formatting of output data(post tagging) needs to be present in order to ensure higher success rate of the manual work being put. Keeping these as our requirements, we built a *Text Annotation Tool* based on *Django* framework which allows us to upload and tag datasets manually. It supports upto 26 different entity labelling and allows for simultaneous collaboration. Finally, it gives us a graphical user interface(GUI) for tagging datasets and helps us automatically format the manually tagged dataset into our desired format. However, it is imperative to decide on a standard format

on which the manually tagged training data will be based on.

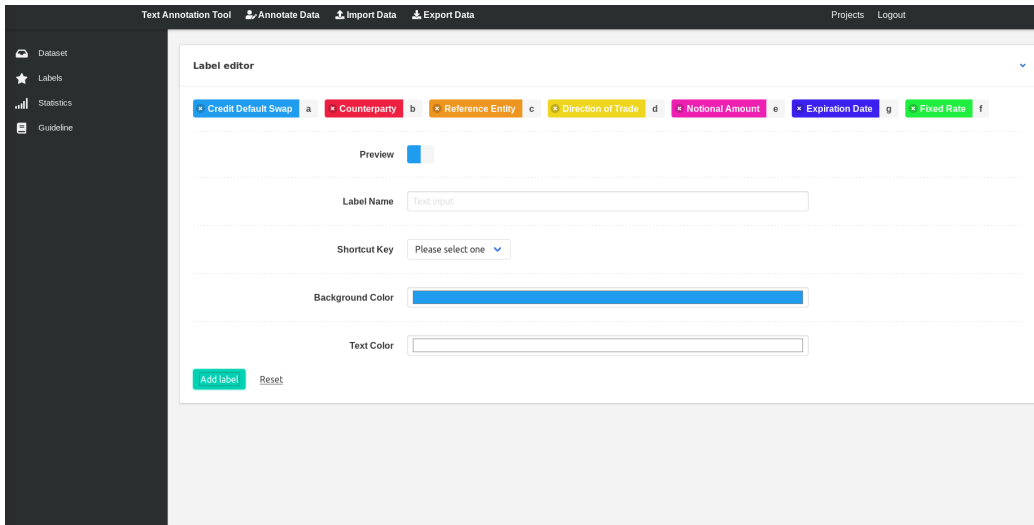


Figure 8: Defining Labels in the Tool

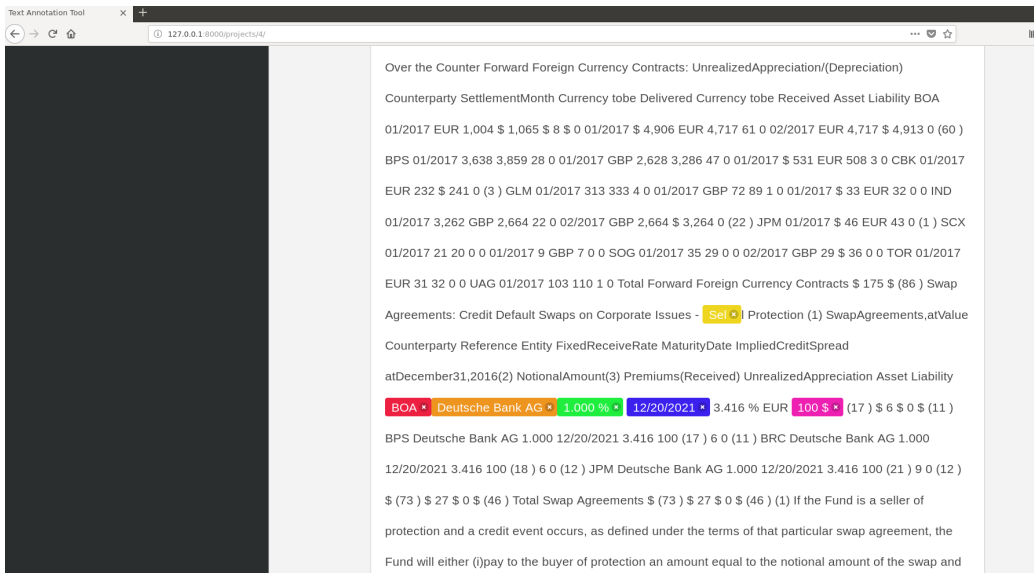


Figure 9: Tagging in Action

Furthermore, we decided to stick with the CoNLL 2003 corpus format which is a standard format that is used in pattern recognition and sequence labelling tasks.

U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

Figure 10: An example of tagged sentence in CoNLL 2003 format with NER(Named Entity Recognition) tags

Upon, successfully tagging a report, you can export the final tagged report to contain sentences in either a .csv or .json format as shown in Figure 9.

### 3.2 Natural Language Processing

A bidirectional LSTM model can be used to take into account an infinite amount of context on both sides of the word. This would then solve the common problem faced by normal feed-forward models which is that of a limited context and this is especially useful for Named Entity Recognition.

The BiLSTM layer has as its input a sequence of word representations (vectors) for the tokens of the input sentence, denoted as  $(y_1, y_2, \dots, y_n)$ . The output of this layer will be a . A sequence of hidden states for each input word vectors,



denoted as  $(h_1, h_2, \dots, h_n)$  will be the output of this BiLSTM layer. The concatenation of the forward  $h_i$  and backward  $h_i$  hidden states will be each final hidden state.

$$\begin{aligned} \overleftarrow{h}_i &= \text{lstm}(\mathbf{x}_i, \overleftarrow{h}_{i-1}), \overrightarrow{h}_i = \text{lstm}(\mathbf{x}_i, \overrightarrow{h}_{i+1}) \\ \mathbf{h}_i &= \left[ \overleftarrow{h}_i ; \overrightarrow{h}_i \right] \end{aligned}$$

Figure 11: Mathematical representation of BiLSTM Layer

It is important to consider the correlations between the neighboring labels and the current labels as there will be many syntactic constraints in natural language sentences. One of the most popular ways to control the structure prediction is Linear-chain Conditional Random Field. It uses a series of potential functions to approximate the conditional probability of the output sequence given the input word sequence.

This means that our input to the CRF layer will be the above sequence of the hidden states and the output will be our final prediction label sequence  $y = (y_1, y_2, \dots, y_n)$  where  $y_i$  denotes set of all the possible labels. We then take  $\mathcal{Y}(\mathbf{h})$  to be the set of all possible label sequences and derive the conditional probability of the output sequence given the input hidden state sequence as

$$p(\mathbf{y}|\mathbf{h}; \mathbf{W}, \mathbf{b}) = \frac{\prod_{i=1}^n \exp(\mathbf{W}_{y_{i-1}, y_i}^T \mathbf{h} + \mathbf{b}_{y_{i-1}, y_i})}{\sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{h})} \prod_{i=1}^n \exp(\mathbf{W}_{y'_{i-1}, y'_i}^T \mathbf{h} + \mathbf{b}_{y'_{i-1}, y'_i})}$$

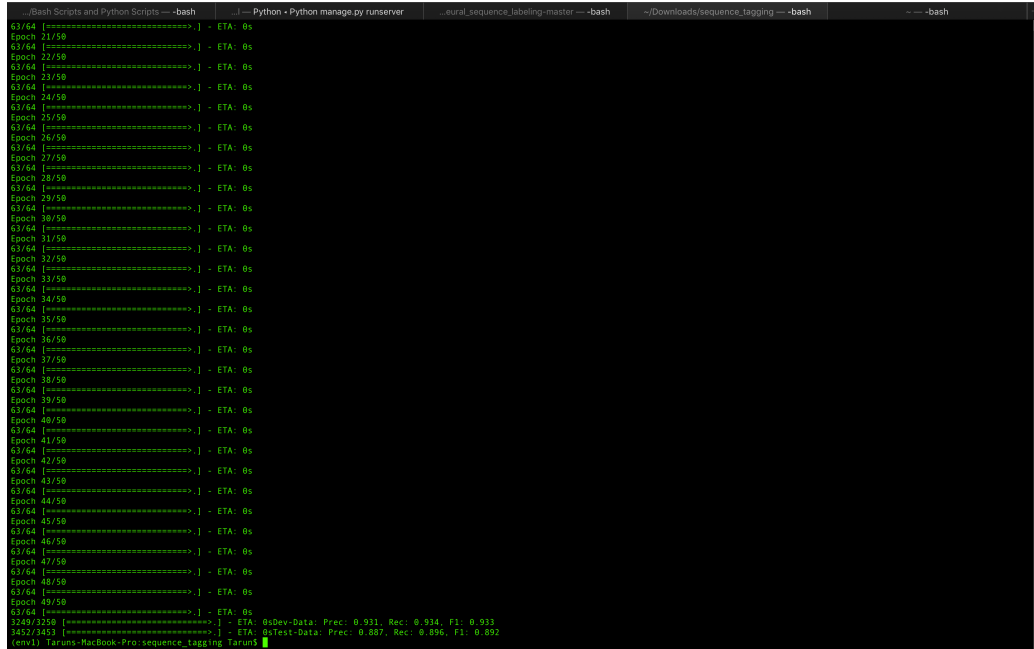
Figure 12: Conditional probability of output sequence

where  $\mathbf{b}$ ,  $\mathbf{W}$  are 2 weight matrices and subscription indicates that we extract the weight vector for the given label pair  $(y_{i-1}, y_i)$ .

## 4 Current Status and Results

We have made considerable progress in our project this semester. We have extracted the information from the structured reports and segregated all the reports into structured and unstructured folders. Within the unstructured folders, we have removed those files which do not contain information relating to Credit Default Swaps. We hence have a folder with the exact reports that we need to extract information from and have reduced the size of the dataset by more than half from 146 GB to around 72 GB on which we need to implement our NLP model. To implement the NLP algorithms, we first have to manually go through all the reports and tag each report with the useful information. We have manually tagged reports in the N-CSR folder using our Text Annotation Tool and then use that as training data for our NLP Model.

We are also underway of developing a document search engine based on *ElasticSearch* which would allow us to aggregate reports with similar style of Credit Default Swap reporting and would enable us to tag those reports swiftly. Given that we are predominantly trying to solve sequence tagging problem Named Entity Relationship tasks, it was important to check whether the implementation that we have done is rightly verified. Therefore, we decided to test our implementation of Bidirectional LSTM-CRF on CoNLL 2003 dataset which is considered as a standard testing methodology in all NER related tasks. We have successfully managed to achieve a very modest *F1* score of 0.892 on the test data while training the model over 50 epochs.



```
03/64 [.....] - ETA: 0s
epoch 21/50
03/64 [.....] - ETA: 0s
epoch 22/50
03/64 [.....] - ETA: 0s
epoch 23/50
03/64 [.....] - ETA: 0s
epoch 24/50
03/64 [.....] - ETA: 0s
epoch 25/50
03/64 [.....] - ETA: 0s
epoch 26/50
03/64 [.....] - ETA: 0s
epoch 27/50
03/64 [.....] - ETA: 0s
epoch 28/50
03/64 [.....] - ETA: 0s
epoch 29/50
03/64 [.....] - ETA: 0s
epoch 30/50
03/64 [.....] - ETA: 0s
epoch 31/50
03/64 [.....] - ETA: 0s
epoch 32/50
03/64 [.....] - ETA: 0s
epoch 33/50
03/64 [.....] - ETA: 0s
epoch 34/50
03/64 [.....] - ETA: 0s
epoch 35/50
03/64 [.....] - ETA: 0s
epoch 36/50
03/64 [.....] - ETA: 0s
epoch 37/50
03/64 [.....] - ETA: 0s
epoch 38/50
03/64 [.....] - ETA: 0s
epoch 39/50
03/64 [.....] - ETA: 0s
epoch 40/50
03/64 [.....] - ETA: 0s
epoch 41/50
03/64 [.....] - ETA: 0s
epoch 42/50
03/64 [.....] - ETA: 0s
epoch 43/50
03/64 [.....] - ETA: 0s
epoch 44/50
03/64 [.....] - ETA: 0s
epoch 45/50
03/64 [.....] - ETA: 0s
epoch 46/50
03/64 [.....] - ETA: 0s
epoch 47/50
03/64 [.....] - ETA: 0s
epoch 48/50
03/64 [.....] - ETA: 0s
epoch 49/50
03/64 [.....] - ETA: 0s
1452/3453 [.....] - ETA: 0sDev-Data: Prec: 0.931, Rec: 0.934, F1: 0.933
1452/3453 [.....] - ETA: 0sTest-Data: Prec: 0.897, Rec: 0.896, F1: 0.892
ven@Taruins-MacBook-Pro:~/sequence_tagging$ tarun@
```

Figure 13: F1, Recall and Precision score on Train and Test Data

## 5 Difficulties Encountered

We encountered challenges in cleaning the data initially due to the volume of data that was present. We had to figure out how to restructure the initial data set and split it in order to proceed with the project in a sequential manner. There was also a lot of data that we didnt need and it was a challenge to identify this data and remove it in order to make our processing much faster. Once this was done, we faced a challenge for tagging the data as we had to go through all the reports manually and find the relevant information. We built our Text Annotation Tool to make this job easier for us.

We also had challenges in setting up our development environment which included two high performance virtual machine. For instance, our virtual machine did not have the ability to render a graphical user interface which hindered our initial progress. However, we used X11 Forwarding option within

SSH which allowed us to view graphical applications on the virtual machine as a workaround.

## **6 Future Plan**

The project follows agile development project management style which allows us to quickly test out new things that we build and fix the issues instantaneously. A detailed plan of action and contingency plan can be found in subsequent parts.

Our immediate plan of action includes completing the document search engine which would allow us to find similar documents in the corpus that we have compiled. Furthermore, combined with our text annotation tool, this would allow us to be even more efficient and accurate with preparing the training data as it would enable us to quickly find reports with a similar style of reporting.

Our next phase of project includes conducting a downstream analysis on the extracted information. This analysis includes objectives like; whether the mutual funds which have complex style of reporting tend to have better returns than the ones with tabulated style or explore the pattern of using index CDS, sovereign CDS, and single-name corporate CDS.

## **7 Conclusion**

Summarizing the report, we show the importance of Credit Default Swaps(CDS) and the power that it has on the economy of a country. There is currently no easy way to access the required information about CDS from the web. We thus

highlight the need for a centralized database consisting of all the information regarding CDS dealings and how analysing this vast amounts of data could provide us with valuable information. We demonstrate the process that can be used to do this which would start with using Python and Perl scripts to get the data. Once we get the data, we preprocess the data to get to the actual useful data and also divide it into two folders, namely the unstructured and structured folder. The reports in the structured folder have their information successfully extracted by the Perl script and hence, we dont need to use them anymore for the tagging process. We focus on the unstructured folder data and clean it more to get the files with the CDS information and then use those files for tagging purposes to extract the data. Furthermore, the power of Natural Language Processing is being used to better understand the financial reports which have an enormous amount of valuable financial insight but from which it is very difficult and time-consuming to extracted with rule-based methods. By exploring deep learning architectures like Bidirectional Long Short Term Memory - Conditional Random Field, we can train machines to help us extract the financial information hidden in the complex methodologies of reporting.

## References

A.Guettler ,T.Adam. Pitfalls and Perils of Financial Innovation: The Use of CDS by Corporate Bond Funds. Jan 10, 2015.

Bill Y.Lin, Frnk F.Xu, Zhiyi Luo and Kenny Q.Zhu, Multi-channel  
"BiLSTM-CRF Model for Emerging Named Entity Recognition in Social  
Media, Shanghai, Sept 2017

B. Shivam. (2018, March 26). Language Modelling and Text Generation using LSTMs- Deep Learning for NLP.

C. Olah (2015, August 27). Understanding LSTM Networks.Available at  
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Credit Default Swaps (n.d). PIMCO Funds. Available at:  
[https://global.pimco.com/en-gbl/resources/education/  
understanding-credit-default-swaps](https://global.pimco.com/en-gbl/resources/education/understanding-credit-default-swaps)

Deep Learning: Convolutional Neural Networks in Python.(n.d). Udemy

P. Wayne (2018, October 6). Credit Default Swaps: An Introduction [Online]. Available: <https://www.investopedia.com/articles/optioninvestor/08/cds.asp>

Susmitha Wunnava, Xiao Qin, Tabassum Kakar, Elke A. Rundensteiner and Xiangnan Kong, *Bidirectional LSTM-CRF for Adverse Drug Event Tagging in Electronic Health Records*, Worcester, 2018

W.Jiang, Z.Zhu. Mutual Funds Holdings of Credit Default Swaps. September 2016.

Zhiheng Huang, Wei Xu, and Kai Yu, *Bidirectional LSTM-CRF Models for Sequence Tagging*, August 2015