INTERNSHIP REPORT

A thesis Submitted in partial

fulfillment of the requirements for

the degree of Master of Technology

in

Data Science (Business

Analytics) By

Deshmukh Mohammad Farooque Khan

(SAP ID: 70271019015)

under the guidance of

Mr. Tushar Kamble (Industry

Mentor) & Prof. Sarada

Samantaray

Department of Data Science

NMIMS University

Mumbai

May, 2021

CERTIFICATE

This is to certify that the thesis entitled **‘Smart AI Data Extraction’** is a bonafide work of **Deshmukh Mohammad Farooque Khan (SAP ID: 70271019015)** submitted to the NMIMS University in the partial fulfillment of the requirement for the award of the degree of ‘**Masters Of Technology**’ in ‘**Data Science (Business Analytics)**’.

Professor Saraday Samantaray (Associate Dean)

NMIMS University

ACKNOWLEDGEMENT

The internship opportunity I had with CRISIL Limited was a great platform for learning about the financial industry. I am very grateful that I could work with such a reputed and employee friendly organization. Also, I got an opportunity to meet professionals who were helpful and led us through this internship.

The Project was conducted under the guidance of my individual external mentors My Manager and Team Members. I thank them for supporting me and providing me with valuable insights throughout the journey of my internship in spite of their busy schedule.

I am immensely grateful to each department of CRISIL that gave me projects, for providing me with the necessary infrastructure and co-operating with me during our tenure. I express my sincere thanks to **Prof. Sarada Samantaray (HOD, Data Science)** for his guidance and for providing the necessary facilities in the department. I am also grateful to the faculty of MTech Data Science Department of Mukesh Patel School Of Technology Management And Engineering, NMIMS for their support whenever needed.

Lastly, I would like to thank my parents for always supporting and encouraging me with their best wishes.

ABSTRACT

Financial analytics and credit rating firms deal with huge amounts of data daily. They provide a vast variety of services, like ratings, data, research, analytics, and advisory solutions to their clients, as well as the government.

Most of the times, these services are provided on a daily/weekly/monthly basis and the manual tasks have to be repeated each time. This calls for a need of integrating the business processes with the automation techniques, so as to optimise human resource. Intelligent automation leads to both reduction of costs and increase of revenue. Hence, integration of automation and analytics in the financial sector has become the need of the hour.

The purpose of this study is to overcome the traditional business analysis approach, and move towards a new-age "man plus machine" approach, in which the complicated and repeated tasks are automated and are easily available through user-friendly platforms. Also, machine learning has been integrated to solve typical problems of the industry.

The study explains, in detail, the importance of introducing automation of Data Extraction in Finance Sector to streamline processes and ensure that everything runs as smoothly as possible.

Key Terms: *Object Detection, Image to Text, Text Classification, Tokenization, Transformers, XLNET, CNN, Finance, Analytics, Predictions.*

TABLE OF CONTENTS

CHAPTER 1: Industry Overview

Analytics in Financial Sector

- The Financial Services Industry is a section of the economy made up of firms and institutions that provide financial services to commercial and retail customers.
- The Financial Services Industry is comprised of a variety of businesses providing services broadly related to insurance, accounting, banking, brokerage, real estate, risk analysis, asset management, and investment.
- Banks and other financial institutions including insurance sector are taking a proactive approach which includes introduction of big data analytics in their industry.
- This has led to a need of firms that provide analytics services to these financial institutions.

Financial Analytics- India Landscape

- Analytics, data science and big data industry in India is currently estimated to be $2.71 billion annually in revenues, growing at a healthy rate of 33.5% CAGR.
- Of the annual inflow to analytics industry, almost 11% can be attributed to advance analytics, predictive modelling and data science.
- A sizeable 22% can be attributed to big data.
- Analytics, data science and big data industry in India is expected to grow seven times in the next seven years. It is estimated to become a 20-billion dollar industry in India by 2025.

ABOUT CRISIL:

CRISIL (formerly Credit Rating Information Services of India Limited) is a global analytical company providing ratings, research, and risk and policy advisory services. CRISIL's majority shareholder is Standard & Poor's, a division of McGraw Hill Financial and provider of financial market intelligence.

CRISIL Services:

CRISIL, India's first credit rating agency, was incorporated in January 1987. The services that the company provides can be broadly described as follows:

- CRISIL is a pioneering credit rating agency in India with ratings on 25,000+ large and mid-sized companies
- SME Ratings and Assessments has made credit accessible to more than 100,000 SMEs in India
- It is India's pre-eminent, independent research house working with 1,000+ Indian and global clients and maintains its own research website
- It is a global leader in high-end research services and risk analytics, and serves top 75 global financial institutions
- CRISIL is a provider of intelligence and analytics to investment banks globally
- CRISIL has helped in shaping policies and frameworks across the infrastructure development cycle in India and other emerging countries
- CRISIL provides a comprehensive range of risk management tools, analytics and solutions
- It provides analytical, research and data services to S&P Global- its major stakeholder

DEPARTMENT OVERVIEW

Corporate Technology

- The Corporate Technology Department caters to the internal CRISIL businesses and also clients. The main motto of the corporate technology department is to integrate technology with analytics.
- The data science team is also a part of the Corporate Technology Department.

- The data sciences solution focuses on solving business problems using data modelling and statistical techniques. They integrate the same with the reporting & visualization framework for ease of dissemination, interpretation and decision making. In addition, there’s the self-service feature for greater end-use control.
- They also deal with risk solutions. They provide a comprehensive range of risk management tools, analytics and solutions to financial institutions, banks and corporates in India, the Middle East, Africa, South Asia and South-East Asia.
- The solutions help clients identify, measure, and calibrate a comprehensive range of risks: credit, price, market, exchange and liquidity, operational, strategic and regulatory.
- They supplement their core consulting strength and analytical skills with robust proprietary software to provide efficient solutions for risk management.

CHAPTER 2: Task 1

Data we received was in the form of various companies annual reports in the form of unstructured data mostly in PDF format. Within these PDF’s there were following list of information:

1. Paragraphs
2. Tables

Now our work at the initial stage was to extract this information from these PDF’s. So first these PDF’s were converted to images. This task was achieved by using Python libraries like PyPDF2, Image, Wand, cv2 etc.

Problems faced while converting these images :

- PDF’s were not properly formatted.
- PDF’s encrypted.
- Showed errors like Expected Object ID’s mismatching with Actual Object ID’s.
- Cross Reference Table is note at the right position.
- PDF’s didn’t open in normal PDF Readers.
- Not all the Images were getting converted.

Why was this necessary ?

Data couldn’t directly be extracted from PDF’s. First the PDF’s need to be converted in the form of Images where each Page of the PDF is converted in the form of Images. Once we receive the Images we can extract information from it.

CHAPTER 3: Task 2

Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularizer and helps reduce overfitting when training a machine learning model. Common Data Augmentation Techniques are listed below:

- Position augmentation
    - Scaling
    - Cropping
    - Flipping
    - Padding
    - Rotation
    - Translation
    - Affine transformation
- Color augmentation
    - Brightness
    - Contrast
    - Saturation
    - Hue

Sub Task 1:
The algorithm would take input as images and their respective XML files and make as many copies of them as designated by the user Using the io library in Python The algorithm saves copies of the images and rename them accordingly. With the XML file we also have to change the text within the file which includes saving the contents of XML file as a string later on after making necessary changes saving them back in .xml format. This algorithm is to be used for data augmentation.

Sub Task 2:
Developed a python code which takes a data frame consisting of Address of the Image and other features of the Image. My task was to write a code which checks if the Image is actually present in the specified folder. If the Images are not present in the specified folder then dump those similar kind of data rows to another dataframe.

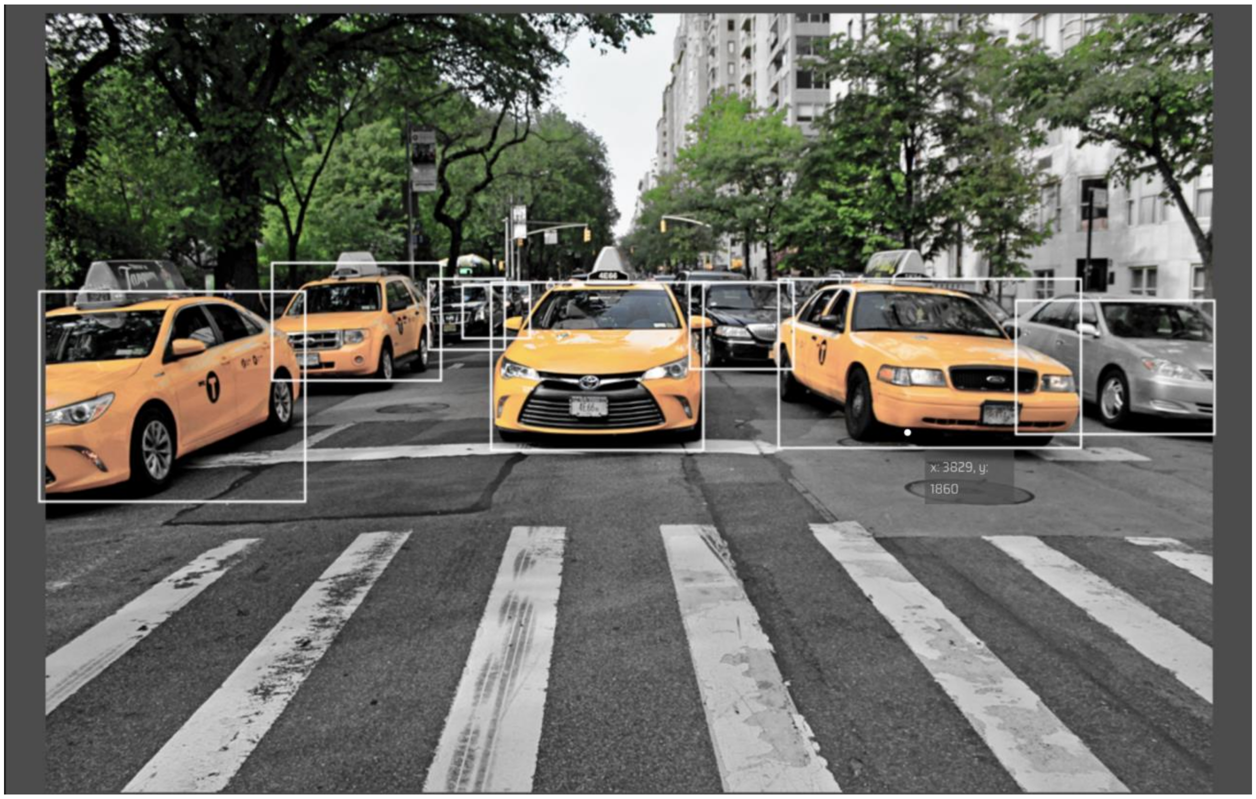
CHAPTER 4: Task 3

Data labelling is an essential step in a supervised machine learning task. **Garbage In Garbage Out** is a phrase commonly used in the machine learning community, which means that the quality of the training data determines the quality of the model. The same is true for annotations used for data labelling. If you show a child a tomato and say its a potato, the next time the child sees a tomato, it is very likely that he classifies it as a potato. As a machine learning model learns in a similar way, by looking at examples, the result of the model depends on the labels we feed in during its training phase.

There are several type of annotation types. Some are listed below:

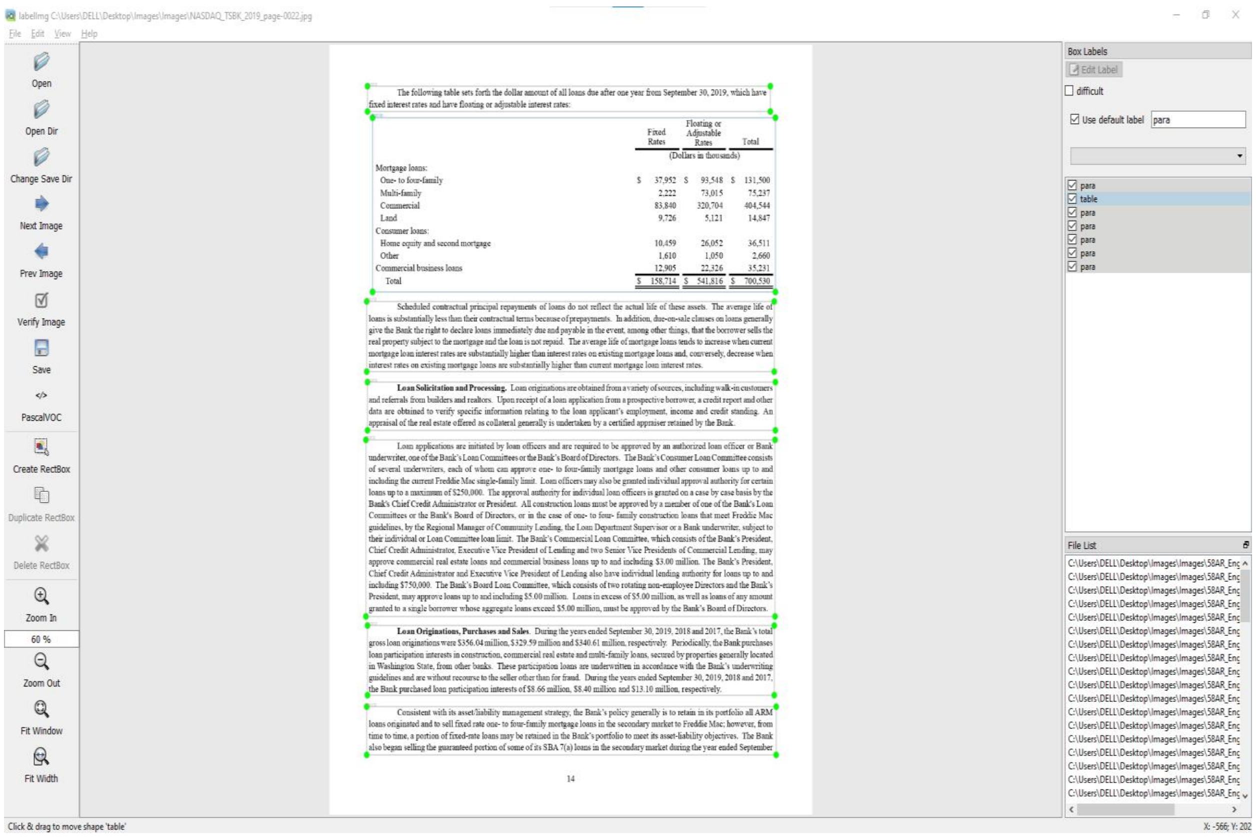
1. Bounding Boxes
2. Polygonal Segmentation
3. Semantic Segmentation
4. 3D Cuboids
5. Key-Point and Landmarks
6. Lines and Splines

Now depending on the use case we use different types of Annotation. Our use cases were based on using Bounding Boxes.



Bounding Boxes: Bounding boxes are the most commonly used type of annotation in computer vision. Bounding boxes are rectangular boxes used to define the location of the target object. They can be determined by the x and y axis coordinates in the upper-left corner and the x and y axis coordinates in the lower-right corner of the rectangle. Bounding boxes are generally used in object detection and localisation tasks. Bounding boxes are usually represented by either two coordinates (x1, y1) and (x2, y2) or by one co-ordinate (x1, y1) and width (w) and height (h) of the bounding box.

Now there are different Annotation tools in the market. We have been using LabelImg which has given us quite good results and is quite easy to use. Below is a snapshot of the tool:



labelImg C:\Users\DELL\Desktop\Images\Images\NASDAQ_TSBK_2019_page-0022.jpg

☐ difficult

☑ Use default label para

☑ para
☑ table
☑ para
☑ para
☑ para
☑ para
☑ para

C:\Users\DELL\Desktop\Images\Images\58AR_Enc

The following table sets forth the dollar amount of all loans due after one year from September 30, 2019, which have fixed interest rates and have floating or adjustable interest rates:

	Fixed Rates	Floating or Adjustable Rates	Total
	(Dollars in thousands)		
Mortgage loans:			
One- to four-family	$ 37,952	$ 93,548	$ 131,500
Multi-family	2,222	73,015	75,237
Commercial	83,840	320,704	404,544
Land	9,726	5,121	14,847
Consumer loans:			
Home equity and second mortgage	10,459	26,052	36,511
Other	1,610	1,050	2,660
Commercial business loans	12,905	22,326	35,231
Total	$ 158,714	$ 541,816	$ 700,530

Scheduled contractual principal repayments of loans do not reflect the actual life of these assets. The average life of loans is substantially less than their contractual terms because of prepayments. In addition, due-on-sale clauses on loans generally give the Bank the right to declare loans immediately due and payable in the event, among other things, that the borrower sells the real property subject to the mortgage and the loan is not repaid. The average life of mortgage loans tends to increase when current mortgage loan interest rates are substantially higher than interest rates on existing mortgage loans and, conversely, decrease when interest rates on existing mortgage loans are substantially higher than current mortgage loan interest rates.

Loan Solicitation and Processing. Loan originations are obtained from a variety of sources, including walk-in customers and referrals from builders and realtors. Upon receipt of a loan application from a prospective borrower, a credit report and other data are obtained to verify specific information relating to the loan applicant's employment, income and credit standing. An appraisal of the real estate offered as collateral generally is undertaken by a certified appraiser retained by the Bank.

Loan applications are initiated by loan officers and are required to be approved by an authorized loan officer or Bank underwriter, one of the Bank's Loan Committees or the Bank's Board of Directors. The Bank's Consumer Loan Committee consists of several underwriters, each of whom can approve one- to four-family mortgage loans and other consumer loans up to and including the current Freddie Mac single-family limit. Loan officers may also be granted individual approval authority for certain loans up to a maximum of $250,000. The approval authority for individual loan officers is granted on a case by case basis by the Bank's Chief Credit Administrator or President. All construction loans must be approved by a member of one of the Bank's Loan Committees or the Bank's Board of Directors, or in the case of one- to four- family construction loans that meet Freddie Mac guidelines, by the Regional Manager of Community Lending, the Loan Department Supervisor or a Bank underwriter, subject to their individual or Loan Committee loan limit. The Bank's Commercial Loan Committee, which consists of the Bank's President, Chief Credit Administrator, Executive Vice President of Lending and two Senior Vice Presidents of Commercial Lending, may approve commercial real estate loans and commercial business loans up to and including $3.00 million. The Bank's President, Chief Credit Administrator and Executive Vice President of Lending also have individual lending authority for loans up to and including $750,000. The Bank's Board Loan Committee, which consists of two rotating non-employee Directors and the Bank's President, may approve loans up to and including $5.00 million. Loans in excess of $5.00 million, as well as loans of any amount granted to a single borrower whose aggregate loans exceed $5.00 million, must be approved by the Bank's Board of Directors.

Loan Originations, Purchases and Sales. During the years ended September 30, 2019, 2018 and 2017, the Bank's total gross loan originations were $356.04 million, $329.59 million and $340.61 million, respectively. Periodically, the Bank purchases loan participation interests in construction, commercial real estate and multi-family loans, secured by properties generally located in Washington State, from other banks. These participation loans are underwritten in accordance with the Bank's underwriting guidelines and are without recourse to the seller other than for fraud. During the years ended September 30, 2019, 2018 and 2017, the Bank purchased loan participation interests of $8.66 million, $8.40 million and $13.10 million, respectively.

Consistent with its asset/liability management strategy, the Bank's policy generally is to retain in its portfolio all ARM loans originated and to sell fixed rate one- to four-family mortgage loans in the secondary market to Freddie Mac; however, from time to time, a portion of fixed-rate loans may be retained in the Bank's portfolio to meet its asset-liability objectives. The Bank also began selling the guaranteed portion of some of its SBA 7(a) loans in the secondary market during the year ended September

14

Click & drag to move shape 'table'

X: -566; Y: 202

Image Annotation Formats:

There is no single standard format when it comes to image annotation. Below are few commonly used annotation formats:

1. Pascal VOC: Pascal VOC stores annotation in XML file. Below is an example of Pascal VOC annotation file for object detection.

```
<annotation>
```

```
 <folder>Train</folder>
```

```xml
<filename>01.png</filename>
<path>/path/Train/01.png</path>
<source>
  <database>Unknown</database>
</source>
<size>
  <width>224</width>
  <height>224</height>
  <depth>3</depth>
</size>
<segmented>0</segmented>
<object>
  <name>36</name>
  <pose>Frontal</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  <occluded>0</occluded>
  <bndbox>
    <xmin>90</xmin>
    <xmax>190</xmax>
    <ymin>54</ymin>
    <ymax>70</ymax>
  </bndbox>
</object>
</annotation>
```

2. YOLO:

In YOLO labeling format, a `.txt` file with the same name is created for each image file in the same directory. Each `.txt` file contains the annotations for the corresponding image file, that is object class, object coordinates, height and width.

```
<object-class> <x> <y> <width> <height>
```

For each object, a new line is created.Below is an example of annotation in YOLO format where the image contains two different objects.

```
0 45 55 29 67
```

```
1 99 83 28 44
```

Now based on the documents we annotated these images. These annotations gave the model basically a location at where are the different types of features located in the Financial Documents which were in the form of Images. Taking an example for a human to find paragraph in the document is easy but for the model to locate this paragraph might be difficult until and unless we train the model feeding them these annotations which include the locations of the features.

Once these annotations were extracted from the Images. We created Ground Truth for Images. We used PyTesseract to convert Image to Text. The annotations were extracted from Images and converted to Text. since the accuracy of the Pytesseract was not as required. We verified each and Text which was saved in an Excel sheet.

Problems Faced while creating to Ground Truth:

Pytesseract was immune to errands when special characters like !, -, $, &, etc. came in the Images. It converted special characters wrongly.

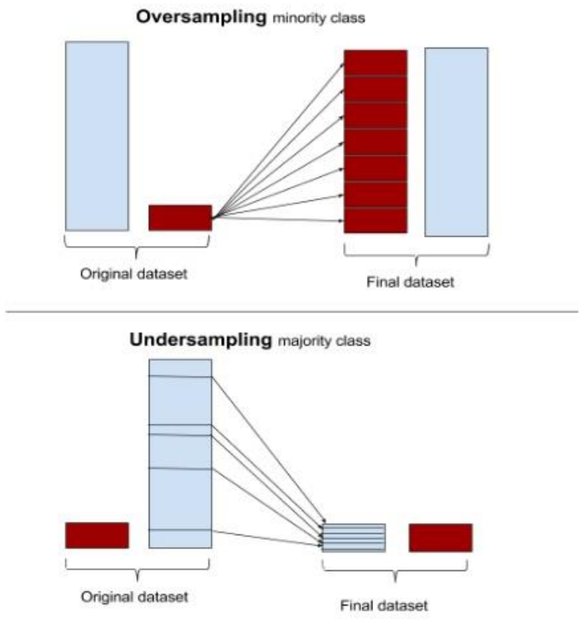
CHAPTER 5: Task 4

The data had irrelevant special characters in between and after each value in the dataframe. Obviously it had to be cleaned before sending this to the model. So, I had written a Python code to clean this type of data where I had split each and every value in the dataframe based on special characters, and extracted the values before it as our original text for the model to read.

The labels columns also had extra unnecessary text before it. Had cleaned our labels column with the same method using .split method of a dataframe. Once the cleaning was done we had the data but one problem was waiting for us.

Oversampling:

The Imbalanced classification problem is what we face when there is a severe skew in the class distribution of our training data. Okay, the skew may not be extremely severe (it can vary), but the reason we identify imbalanced classification as a problem is because it can influence the performance on our Deep Learning algorithms.


Oversampling minority class
Original dataset
Final dataset
Undersampling majority class
Original dataset
Final dataset

One way the imbalance may affect our Deep Learning algorithm is when our algorithm completely ignores the minority class. The reason this is an issue is because the minority class is often the class that we are most interested in. For instance, when building a classifier to classify fraudulent and non-fraudulent transactions from various observations, the data is likely to have more non-fraudulent transactions than that of fraud — I mean think about it, it would be very worrying if we had an equal amount of fraudulent transactions as non-fraud.

An approach to combat this challenge is *Random Sampling.* There are two main ways to perform random resampling, both of which have there pros and cons.

Oversampling — Duplicating samples from the minority class

Undersampling — Deleting samples from the majority class.

In other words, Both oversampling and undersampling involve introducing a bias to select more samples from one class than from another, to compensate for an imbalance that is either already present in the data, or likely to develop if a purely random sample were taken.

Our sample data consisted of 690 data points. There were a total of 47 Labels. The maximum count of one label was 350. The rest 46 labels just repeated 3-4 times. Now a python code was written to oversample the remaining 46 Labels to bring it to the count of the maximum repeated label.

Why Oversampling was done ?

For Deep Learning Algorithms affected by skewed distribution, such as artificial neural networks, this is a highly effective technique. However, tuning the target class distribution is advised in many scenarios as seeking a balanced distribution for a severely imbalanced dataset can lead to the algorithm overfitting the minority class, in turn resulting in an increase of our generalization error. Another thing we ought to be aware of is the increased computational cost. Increasing the number of examples in the minority class (especially for a severely skewed data set) may result in an increased computational when we train our model and considering the model is seeing the same examples multiple times, this isn’t a good thing. Nonetheless, Oversampling is a pretty decent solution and should be tested.

Once the data was oversampled, we then converted the Labels into numbers using the function of tenserflow.keras.utils to_categorical.It basically Label Encodes the string values to numbers.

But why do we Label Encode ?

As you might know by now, we can't have text in our data if we're going to run any kind of model on it. So before we can run a model, we need to make this data ready for the model. And to convert this kind of categorical text data into model-understandable numerical data, we use the Label Encoder class.

We then tokenized the text column using tensorflow.keras.preprocessing.text’s Tokenizer function. Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.

Before processing a natural language, we need to identify the *words* that constitute a string of characters. That’s why tokenization is the most basic step to proceed with NLP (text data). This is important because the meaning of the text could easily be interpreted by analyzing the words present in the text.

Let’s take an example. Consider the below string:

“This is a cat.”

What do you think will happen after we perform tokenization on this string? We get [‘This’, ‘is’, ‘a’, cat’].

Once Tokenization is done, we then need to do padding, since every sentence in the text has not the same number of words, we can also define maximum number of words for each sentence, if a sentence is longer then we can drop some words. The result of the padding sequences is pretty straight forward. We could observe that the list of sentences that have been padded out into a matrix where each row in the matrix has an encoded sentence with the same length this is due to the

- Additional Zeroes for short sentences and
- Truncating the sentences which exceeds the max number of words which is declared by max length.

Embedding:

So far we have words in a sentence and often the sentences have similar meanings e.g. in a sentiment analysis it can be on e.g. on high level negative or positive which can be used to cluster together. Here the meaning of the sentences can come from the labeling of the dataset. The sentences with the same label will have a similar meaning as well as similar vectors. Associating the vectors with the labels is called an embedding i.e., the vectors for each word with their associated sentiment.

When the input to a neural network contains symbolic categorical features (e.g. features that take one of k distinct symbols, such as words from a closed vocabulary), it is common to associate each possible feature value (i.e., each word in the vocabulary) with a d-dimensional vector for some d. These vectors are then considered parameters of the model, and are trained jointly with the other parameters.

Three techniques that can be used to learn a word embedding from text data:

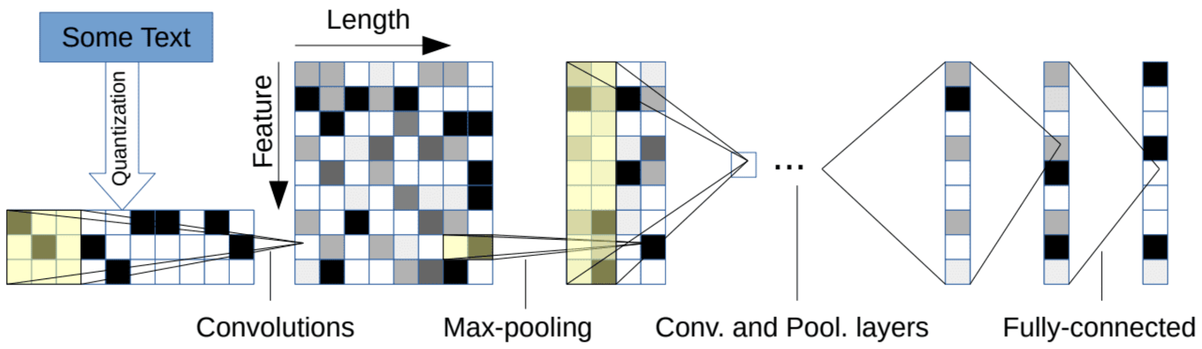
1. Embedding Layer
2. Word2Vec
3. GloVe

According to our use case I used Glove Embeddings.

CHAPTER 6: Task 5

A standard model for text classification is to use an Embedding layer as input, followed by a one-dimensional convolutional neural network, pooling layer, and then a prediction output layer.

Below is the general architecture of a CNN for text classification:



We use the output of the embedding layer applied previously. When we do dot product of vectors representing text, they might turn out zero even when they belong to the same class but when we do dot product of those embedded vectors to find similarity between them then we find interrelation of words for a specific class.

Now we add a sequence of Convolutional layers and Max-Pooling layers. To avoid overfitting we added the use of some regularization techniques such as adding Dropout layers in between. Finally we flatten those matrices into vectors and add dense layers. The last dense layer is having 47 as parameter because we are doing a multilabel classification.

Now we fit the our training data and define the epochs (number of passes through dataset) and batch size(number of samples processed before updating the model) for our learning model. Batch Size is usually kept greater than or equal to 1 and less than the number of samples in the training data.

We were able to achieve an accuracy of 82% over our dataset's test data.

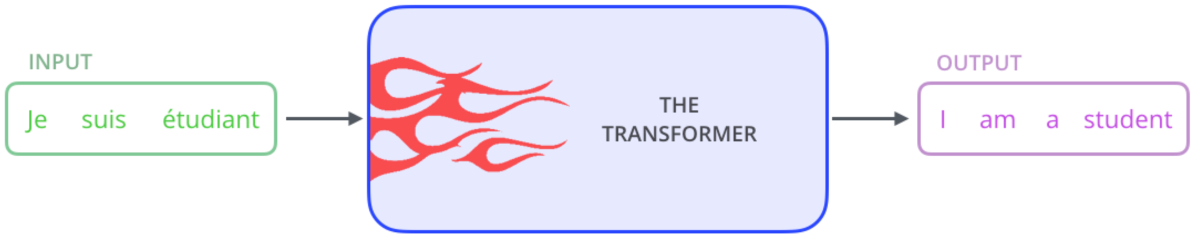
What problems did CNN solve ??

- Sequential computation inhibits parallelization
- No explicit modeling of long and short range dependencies
- "Distance" between positions is linear

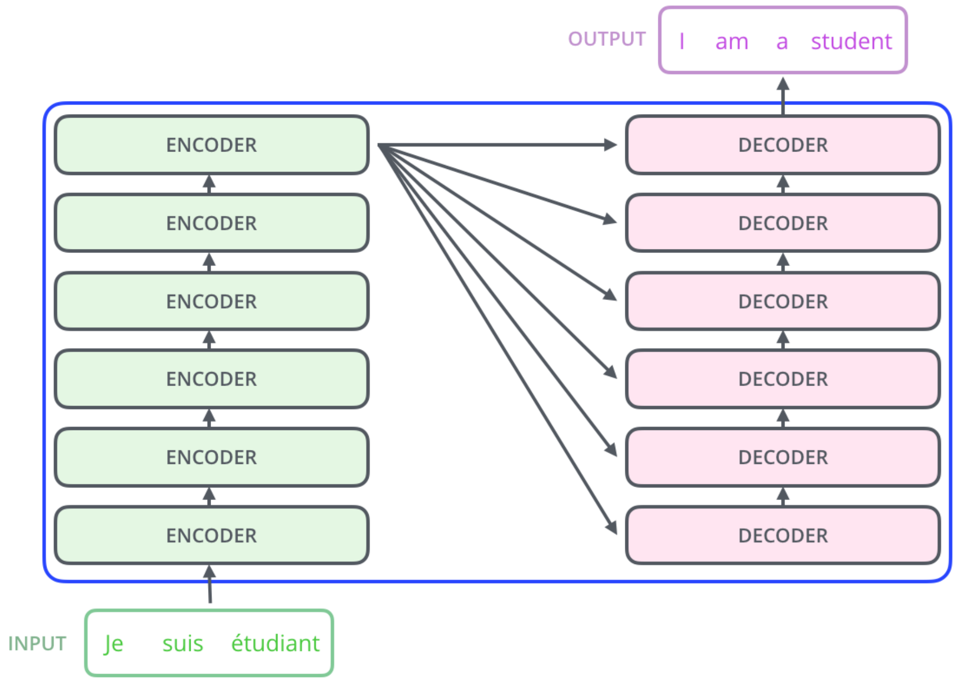
Our CNN's enough ?

The problem is that Convolutional Neural Networks do not necessarily help with the problem of figuring out the problem of dependencies when translating sentences. That's why Transformers were created, they are a combination of both CNNs with attention.

Transformer is a model that uses attention to boost the speed. More specifically, it uses self-attention.



Transformer consists of six encoders and six decoders. Now the number six is just a hyperparameter you can change it as per your hit and trial result.



Each encoder is very similar to each other. All encoders have the same architecture. Decoders share the same property, i.e. they are also very similar to each other. Each encoder consists of two layers: **Self-attention** and a feed Forward Neural Network.

Attention is simply a vector, often the outputs of dense layer using softmax function. Before Attention mechanism, translation relies on reading a complete sentence and compress all information into a fixed-length vector, as you can image, a sentence with hundreds of words represented by several words will surely lead to information loss, inadequate translation, etc. However, attention partially fixes this problem. It allows machine translator to look over all the information the original sentence holds, then generate the proper word according to current word it works on and the context. It can even allow translator to zoom in or out (focus on local or global features).

Why Attention ?

Self-Attention is compression of attentions toward itself. The main advantages of Self-Attention Layer compares to previous architectures are:

1. Ability of parallel computing (compares to RNN)
2. No need of deep network to look for long sentence (compares to CNN)

To speak correctly, Convolutional Neural Network(CNN) based NLP is also solutions for Recursive Neural Network based one. However the disadvantages of CNN are vivid, its not suitable for process long sentence. Self-Attention Layer checks attention with all words in same sentence at once, which makes it a simple matrix calculation and able to parallel computes on computing units. Also, Self-Attention Layer can use Multi-Head architecture mentioned below to broaden the vision (associated word’s distance in sentence).

What is positional encoding and Why do we need it in the first place?
Position and order of words are the essential parts of any language. They define the grammar and thus the actual semantics of a sentence.

We used the State-of-art model for text classification.

When comparing both our models CNN and State-of-art model, **State-of-art model’s** achieved great results in text classification.

CHAPTER 7: Limitations

1. The number of Fields we mapped after the sentences were created were limited to 30.
2. Different Tabular Structures affected our code for Sentence formation.

CHAPTER 8: Future Scopes

1. Number of fields was limited to a certain number but could be extended later on.
2. The **Smart AI Extraction** ca also be used for various different use cases in the Financial Sector.

CHAPTER 9: BIBLIOGRAPHY

1. Attention Is All You Need.
2. Text Classification Based on Convolutional Neural Networks and Word Embedding for Low-Resource Languages: Tigrinya
3. XLNet: Generalized Autoregressive Pre-training for Language Understanding Presented by Andrew Or, Ksenia Sokolova Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le
4. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding