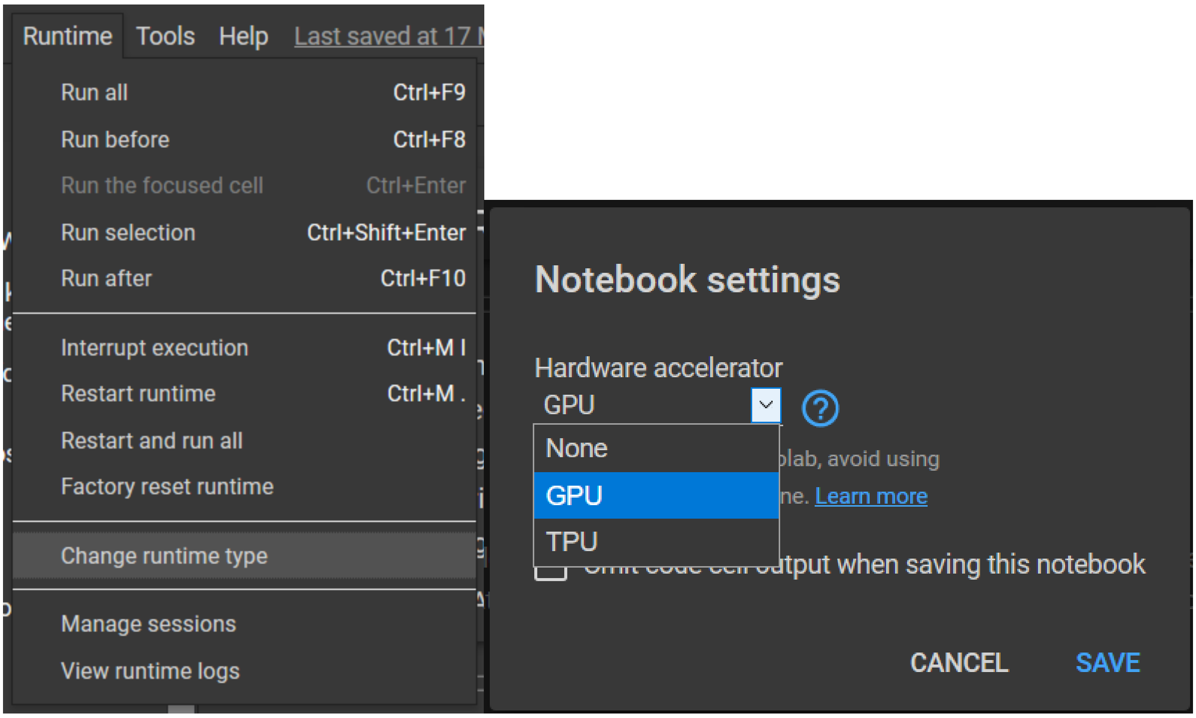
Phase 1 Model Training Documentation

The purpose of this documentation is to demonstrate the steps a user should take to successfully run the ipython file for model training. This documentation is for data scientists who want to see what the process of training the model is, and possibly to use their own data to train the model.

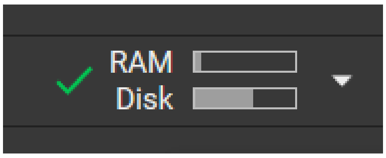
This Model Training documentation is for training a Multilabel MLP Model.

Instructions

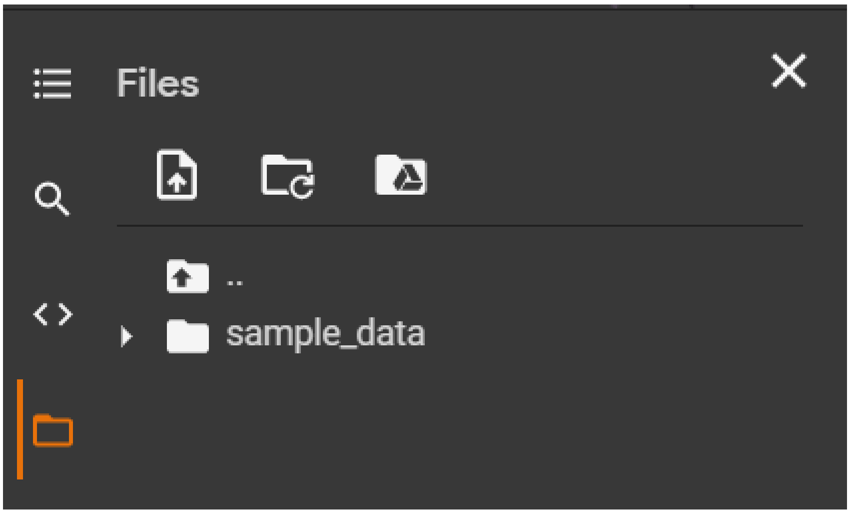
1. Upload "NLP_MLP_Multilabel.ipynb" to Google Colaboratory (Google Colab).
2. Under the Runtime tab, change runtime type to "GPU" and click on "SAVE". This will allow for faster running times when running the notebook so that time can be saved.

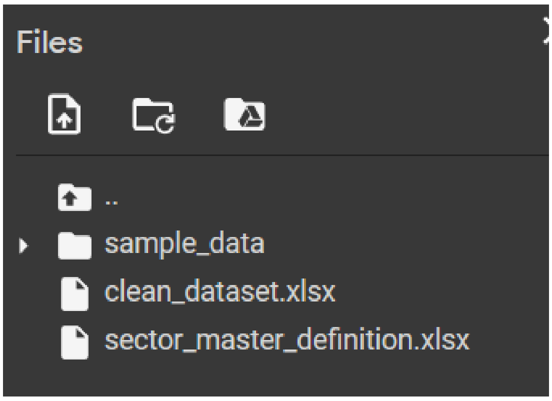


3. Connect to colab runtime by clicking on the connect button found on the top right bar if not already connected. You should see the following if the connection is successful.



4. On the left tab, click on the icon of a folder and upload “clean_dataset.xlsx” and “sector_master_definition.xlsx”





5. Now, the notebook is ready to run. Run the notebook cells one at a time starting from the start of Section 2 (Data Importing) until the end of Section 5 (Models). **Do not run any cell in section 1, as it may cause the training to fail.**

2.Data Importing
- 2.1. Load the libraries
- 2.2. Check CUDA Version
- 2.3. Load the modules
- 2.4. Load the dataset

3.Exploratory Data Analysis
- 3.1. Get overview of dataset
- 3.2. Drop unncessary columns
- 3.3. Filter rows with valid data
- 3.4. Get graphical overview of dataset
- 3.5. See examples of company description

4.Data Preprocessing
- 4.1. Removing \n
- 4.2. Calculating the word length distribution
- 4.3. Subsample from the entire dataset
- 4.4. Populating Nan cells
- 4.5. Assigning tags
- 4.6. Text Tokenization, Removing Stop Words, punctuations, numbers, stop words and Lower Case
- 4.7. Bag of Words / TF-IDF

5.Models
- 5.1. Training Models
- 5.2 Testing Models
- 5.3. Saving Models

6. After running all the cells, the model is now trained.

In-depth descriptions

Here, we install and inspect the various libraries required for the project. The libraries installed are shown below.

```python
# install necessary libraries that might not be found
!pip install -U spacy
!python -m spacy validate
!pip install -U pip setuptools wheel
!pip install -U spacy[cuda110,transformers,lookups]
!python -m spacy download en_core_web_lg

# check versions of libraries we are going to use
%tensorflow_version 2.x
import os
import tensorflow
import sklearn
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib
import spacy
import platform

message="        Versions        "
print("*"*len(message))
print(message)
print("*"*len(message))
print("Tensorflow version={}".format(tensorflow.__version__))
print("Keras version={}".format(tensorflow.keras.__version__))
print("Sklearn version={}".format(sklearn.__version__))
print("Numpy version={}".format(np.__version__))
print("Pandas version={}".format(pd.__version__))
print("Seaborn version={}".format(sns.__version__))
print("Matplotlib version={}".format(matplotlib.__version__))
print("SpaCy version={}".format(spacy.__version__))
print("Python version={}".format(platform.python_version()))
```

The inspected libraries are shown here.

```
************************
        Versions        
************************
Tensorflow version=2.4.1
Keras version=2.4.0
Sklearn version=0.22.2.post1
Numpy version=1.19.5
Pandas version=1.1.5
Seaborn version=0.11.1
Matplotlib version=3.2.2
SpaCy version=3.0.6
Python version=3.7.10
```

Here, we load the libraries into their respective namespaces. We then import the dataset and inspect it.

```
[ ]
# importing necessary modules for this project
import tensorflow as tf

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import string
import spacy

# activate the GPU to run spaCy with GPU
spacy.prefer_gpu()

%matplotlib inline
```

2.4. Load the dataset

Load the dataset for usage in the entire project.

```
[ ]
# use pandas to read the excel file and populate it in a pandas dataframe
companies = pd.read_excel('./clean_dataset.xlsx')

# see the top 10 companies that are populated in the dataframe
companies.head(10)
```

A preview of the dataset is shown here.

	Company_ID	Company	Country	PIC	Sector	Subsector	Archetype	Valuechain	Websites
0	4137190062363536	AKSORN CHAROEN TAT ACT. CO.,LTD.	THAILAND	NaN	TMT	media	media_aggregator/distributor	Midstream	https://getlinks.co/6831
1	23248790229909748	DONGGUAN SHENGYA CLEANING APPLIAME CO.LTD	CHINA	NaN	TMT	consumer electronics	consumer electronics_distributor	Downstream	https://baike.baidu.com/item/%E4%B8%9C%E8%8E%9...
2	28486505934571008	EXIS TECH SDN. BHD.	MALAYSIA	NaN	oos	others	others	NaN	http://www.exis-tech.com/
3	38251695094669872	BEI JING ESTRABA IMPORT AND EXPORT	CHINA	NaN	NaN	NaN	NaN	NaN	NaN
4	39910921263510776	Aztech Electronics Pte Ltd	SINGAPORE	NaN	TMT	consumer electronics	consumer electronics_distributor	Downstream	https://www.aztech.com/business/about-us/
5	54863889264716592	TONGDUN INTERNATIONAL PTE LTD	SINGAPORE	NaN	tmt	it_services	it_services	midstream	https://www.tongdun.net/info/company

Here, we start performing Exploratory Data Analysis on the dataset. Currently, this is printing out the various columns present in the dataset.

```
[ ]  # see the row headers of the entire pandas dataframe first
     list(companies.columns)
```

```
['Company_ID',
 'Company',
 'Country',
 'PIC',
 'Sector',
 'Subsector',
 'Archetype',
 'Valuechain',
 'Websites',
 'Company Profile Information',
 'Remarks']
```

Here, the number of records, as well as the number of unique labels are shown.

```
# get the total number of records in the dataframe
df_count = companies['Company_ID'].count()

# get count of unique contries where companies are based in
df_countCountry = companies['Country'].nunique()

# get count of total unique sectors where companies are from
df_countSector = companies['Sector'].nunique()

# get count of total unique subseector where companies are from
df_countsubSector = companies['Subsector'].nunique()

# get count of total unique valuechain where companies are from
df_countValuechain = companies['Valuechain'].nunique()

print('Total number of records:', df_count)
print('Total number of countries:', df_countCountry)
print('Total number of sectors:', df_countSector)
print('Total number of subsectors:', df_countsubSector)
print('Total number of valuechain:', df_countValuechain)
```

```
Total number of records: 1000
Total number of countries: 7
Total number of sectors: 14
Total number of subsectors: 26
Total number of valuechain: 18
```

Here we print out the unique Archetype labels present in the dataset.

```
print('List of unique archetype:\n{}'.format(df_archetype))
```

```
List of unique archetype:
buildings & industrial_contractor     143
others                                115
consumer electronics_distributor       96
building_material_manufacturer         62
it_services                            59
                                     ...
base metal distributor                  1
infrastructure_sub contractor           1
hotels and accommodation_developer      1
o&g_refiner                             1
palm oil                                1
Name: Archetype, Length: 61, dtype: int64
```

Here, we drop the redundant columns from the dataset, so they do not interfere with the EDA process.

Here, we will drop columns that wiill not aid in our EDA.

```
# declare the list of the row names that are redundant
rows_to_drop = ['Company_ID', 'PIC', 'Websites', 'Remarks']

# use a conditional expression to filter out those rows
df_filteredCompanies = companies.drop(labels=rows_to_drop, axis=1)

df_filteredCompanies
```

	Company	Country	Sector	Subsector	Archetype	Valuechain	Company Profile Information
0	AKSORN CHAROEN TAT ACT. CO.,LTD.	THAILAND	TMT	media	media_aggregator/distributor	Midstream	For over 80 years of experience in creating an...
1	DONGGUAN SHENGYA CLEANING APPLIAME CO.LTD	CHINA	TMT	consumer electronics	consumer electronics_distributor	Downstream	Yatai's main products cover various cleaning m...
2	EXIS TECH SDN. BHD.	MALAYSIA	oos	others	others	NaN	In the beginning, it started off by providing ...
3	BEI JING ESTRABA IMPORT AND EXPORT	CHINA	NaN	NaN	NaN	NaN	NaN
4	Aztech Electronics Pte Ltd	SINGAPORE	TMT	consumer electronics	consumer electronics_distributor	Downstream	Being a turnkey, one-stop integrated solutions...
...	...	...	...	...	...	...	...
995	LUX DISTRIBUTOR SDN. BHD.	MALAYSIA	cni	building_material	building_material_supplier_distributor	downstream	Lux Distributor Sdn Bhd provides building prod...
996	KJI INDUSTRIAL LIMITED	HONG KONG	tmt	consumer electronics	consumer electronics_distributor	downstream	Our wide product range including Citrus Juicer...
997	TEKNOSERV ENGINEERING SDN. BHD.	MALAYSIA	cni	utilities	utilities_developer	upstream	Teknoserv Engineering Sdn Bhd provides water s...
998	THERMAL SOLUTIONS ASIA PTE LTD	SINGAPORE	CNI	cni_service providers	cni_service providers	MIDSTREAM	Founded in 2000, Thermal Solutions Asia Pte Lt...
999	AUDIO ZOOM PTE LTD	SINGAPORE	TMT	it_services	it_services	Midstream	AUDIO ZOOM is an ACRA-registered entity that h...

1000 rows × 7 columns

Records with all NaN values are extracted, shown, and discarded, as they will only make the dataset noisy and less effective.

```
# find all the rows with nan data in sector, subsector, archetype and valuechain
cols_to_check = ['Sector', 'Subsector', 'Archetype', 'Valuechain', 'Company Profile Information']
empty = df_filteredCompanies[df_filteredCompanies[cols_to_check].isnull().all(1)]

empty
```

	Company	Country	Sector	Subsector	Archetype	Valuechain	Company Profile Information
3	BEI JING ESTRABA IMPORT AND EXPORT	CHINA	NaN	NaN	NaN	NaN	NaN
13	CHANGTU COUNTRY LONGXING FERTILIZER CO.,LTD	CHINA	NaN	NaN	NaN	NaN	NaN
35	ZIBOBOSHANHONGLIWEI MOTOR CO.,LTD	CHINA	NaN	NaN	NaN	NaN	NaN
62	BEIJING DUO MEIDUO SHIYOU PRODUCTS SALES CO., ...	CHINA	NaN	NaN	NaN	NaN	NaN
79	TRUSVEST SDN. BHD.	MALAYSIA	NaN	NaN	NaN	NaN	NaN
108	LUOTIAN COUNTY SILICON CARBIDE PLANT	CHINA	NaN	NaN	NaN	NaN	NaN
111	NEWTON INTERNATIONAL (HK) LIMITED	SINGAPORE	NaN	NaN	NaN	NaN	NaN
129	ZHEJIANG-THAI PHOTOVOLTAIC TECHNOLOGY CO., LTD.	CHINA	NaN	NaN	NaN	NaN	NaN
162	RAINBOW BLISS LIMITED	SINGAPORE	NaN	NaN	NaN	NaN	NaN
195	VATCHAREE TANGTRAKOOLCHAROEN	THAILAND	NaN	NaN	NaN	NaN	NaN
220	LIAONING JINYUN ELECTROMECHANICAL EQUIPMENT CO...	CHINA	NaN	NaN	NaN	NaN	NaN
221	NANJING JIASHAN CONCRETE CO., LTD.	CHINA	NaN	NaN	NaN	NaN	NaN

The records with valid data are then kept.

```
# now we get the dataset that are valid
df_valid = pd.concat([df_filteredCompanies, empty, empty]).drop_duplicates(keep=False)

df_valid
```

	Company	Country	Sector	Subsector	Archetype	Valuechain	Company Profile Information
0	AKSORN CHAROEN TAT ACT. CO.,LTD.	THAILAND	TMT	media	media_aggregator/distributor	Midstream	For over 80 years of experience in creating an...
1	DONGGUAN SHENGYA CLEANING APPLIAME CO.LTD	CHINA	TMT	consumer electronics	consumer electronics_distributor	Downstream	Yatai's main products cover various cleaning m...
2	EXIS TECH SDN. BHD.	MALAYSIA	oos	others	others	NaN	In the beginning, it started off by providing ...
4	Aztech Electronics Pte Ltd	SINGAPORE	TMT	consumer electronics	consumer electronics_distributor	Downstream	Being a turnkey, one-stop integrated solutions...
5	TONGDUN INTERNATIONAL PTE LTD	SINGAPORE	tmt	it_services	it_services	midstream	Tongdun Technology is a professional third-par...
...	...	...	...	...	...	...	...
995	LUX DISTRIBUTOR SDN. BHD.	MALAYSIA	cni	building_material	building_material_supplier_distributor	downstream	Lux Distributor Sdn Bhd provides building prod...
996	KJI INDUSTRIAL LIMITED	HONG KONG	tmt	consumer electronics	consumer electronics_distributor	downstream	Our wide product range including Citrus Juicer...
997	TEKNOSERV ENGINEERING SDN. BHD.	MALAYSIA	cni	utilities	utilities_developer	upstream	Teknoserv Engineering Sdn Bhd provides water s...
998	THERMAL SOLUTIONS ASIA PTE LTD	SINGAPORE	CNI	cni_service providers	cni_service providers	MIDSTREAM	Founded in 2000, Thermal Solutions Asia Pte Lt...
999	AUDIO ZOOM PTE LTD	SINGAPORE	TMT	it_services	it_services	Midstream	AUDIO ZOOM is an ACRA-registered entity that h...

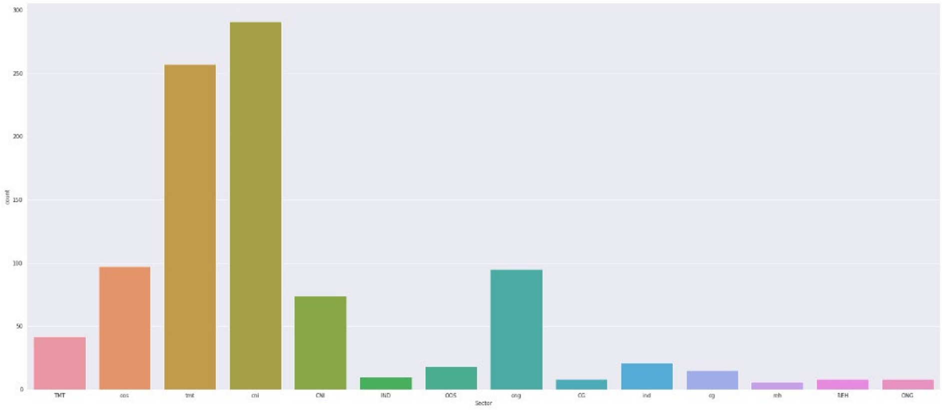
Here, we will turn the numbers seen in the EDA into graphs, so that it is easier to spot trends.

3.4. Get graphical overview of dataset

Get visualised information of the dataset to understand the dataset better.

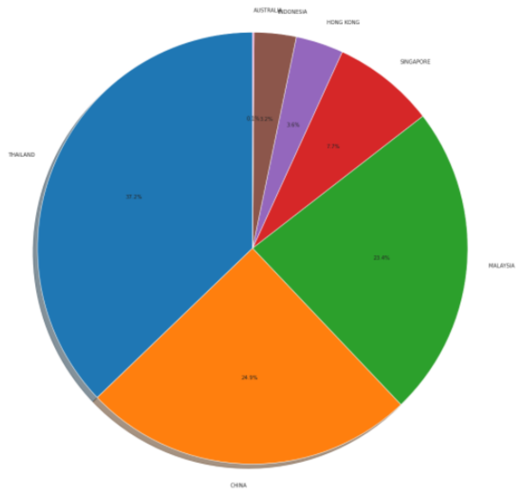
```
sns.set_style('darkgrid')
plt_dims = (30, 13)
fig, ax = plt.subplots(figsize=plt_dims)

# plot a barplot to see number of companies that belongs to specific sectors
sns.countplot(x="Sector", data=df_valid, ax=ax)
plt.show()
```

Shown here is a barchart of the breakdown of labels in Sector. It seems that tmt, cni and reh make up the majority of the dataset.



Here, a piechart of the countries found in the dataset is plotted. It seems that Thai, Chinese and Malaysian companies make up a large majority of the dataset.

```
In [16]: # Pie chart
labels = list(df_valid['Country'].unique())
sizes = list(df_valid['Country'].value_counts())

plt_dims = (30, 13)
fig1, ax1 = plt.subplots(figsize=plt_dims)
ax1.pie(sizes, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.show()
```



Here, a subset of the company profile descriptions is printed, so that we can see an example of a description as well as see how we can do data pre-processing.

```
In [17]: # configurate pandas dataframe to let us see the entire company description IN FULL
         pd.set_option('display.max_colwidth', None)

         # get the 1st 50 results and observe
         df_valid.loc[0:10,'Company Profile Information']
```

```
Out[17]: 0
         For over 80 years of experience in creating and developing high-quality learning materials has enabled us to provide world-class educatio
         nal innovation to meet the needs of all teachers, students, institutions and educational authorities.
         1
         Yatai's main products cover various cleaning machinery, cleaning agents, cleaning tools, stone maintenance and other cleaning products; c
         leaning solutions and services include product technical consultation, product customization, employee training, maintenance and so on.\n
         2
         In the beginning, it started off by providing technical support for test handlers, then moving into module design and production. Its fir
         st, in-house designed, full-fledged handler was introduced in 2008.\nSince then, the company has designed and produced a wide range of tu
         rret and pick-and-place solutions for its customers all over the world.
         4                                                                                            Being a turnkey, one-stop integrated solutions provider based
         in Singapore, Aztech is equipped with state-of-the-art equipment, R&D, design, manufacturing and packaging capabilities to deliver a seam
         less, unified experience. Each and every time. Always striving towards the edge of technology for more than 34 years, we have been buildi
         ng capabilities to serve clients’ manufacturing needs, including the consumer electronics, telecommunications, healthtech, LED lighting,
         automotive and technology start-up market segments.
         5       Tongdun Technology is a professional third-party intelligent risk management and decision-making service provider headquartered in
         Hangzhou, Zhejiang. By integrating artificial intelligence into business scenarios, Tongdun Technology offers solutions in intelligent us
         er analysis, intelligent risk management, intelligent antifraud and intelligent operation to clients from various industries including fi
```

Here, we remove the newline (\n) character from the company descriptions, so that they do not interfere with the tokenisation process in the future.

4.Data Preprocessing

4.1. Removing \n

Now, we will like to standardize all the paragraphs such that the are homogenous, before we tokenize the paragraph.

```
[33] # get rid of the \n found in the respective descriptions
     df_valid = df_valid.replace('\n',' ', regex=True)

     # now we validate to see if theye are really gone
     df_valid.loc[0:10,'Company Profile Information']
```

```
ping high-quality learning materials has enabled us to provide world-class educational innovation to meet the needs of all teachers, students, institutions and educational authorities.
stone maintenance and other cleaning products; cleaning solutions and services include product technical consultation, product customization, employee training, maintenance and so on.
full-fledged handler was introduced in 2008. Since then, the company has designed and produced a wide range of turret and pick-and-place solutions for its customers all over the world.
apabilities to serve clients’ manufacturing needs, including the consumer electronics, telecommunications, healthtech, LED lighting, automotive and technology start-up market segments.
ial industry, internet business, logistics, healthcare, retail, smart cities and government bodies. Over 10,000 corporate clients have chosen Tongdun Technology’s products and services
has been servicing the construction and the oil and gas industries with distinction.We try our hardest to create win-win situations and value-add to our clients with every transaction.
                                                                                                                                                   Sales of hydraulic equipment and parts
                                                                                                                     WHOLESALE OF TELECOMMUNICATIONS EQUIPMENT (EXCLUDING HANDPHONES)
increasing demands of various industries.  The vision became a reality in 1988 when the Excelkos Group was established to distribute and provide quality rubber industrial chemicals.
                        Kum Eng Huat is the authorised dealer for Osram and Philips, as well as a trusted supplier of lighting solutions for major projects with the local government.
```

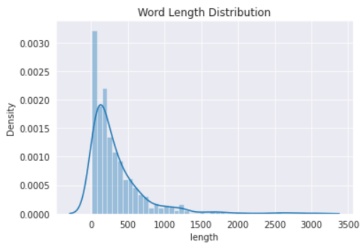
Here, we calculate the world length distribution of each company description.

```python
# first, add in a new column that tabluates the length of the respecive company description
df_valid["length"] = df_valid["Company Profile Information"].str.len()
df_valid.head()
```

	Company	Country	Sector	Subsector	Archetype	Valuechain	Company Profile Information	length
0	AKSORN CHAROEN TAT ACT. CO.,LTD.	THAILAND	TMT	media	media_aggregator/distributor	Midstream	For over 80 years of experience in creating and developing high-quality learning materials has enabled us to provide world-class educational innovation to meet the needs of all teachers, students, institutions and educational authorities.	238.0
1	DONGGUAN SHENGYA CLEANING APPLIAME CO.LTD	CHINA	TMT	consumer electronics	consumer electronics_distributor	Downstream	Yatai's main products cover various cleaning machinery, cleaning agents, cleaning tools, stone maintenance and other cleaning products; cleaning solutions and services include product technical consultation, product customization, employee training, maintenance and so on.	273.0
2	EXIS TECH SDN. BHD.	MALAYSIA	oos	others	others	NaN	In the beginning, it started off by providing technical support for test handlers, then moving into module design and production. Its first, in-house designed, full-fledged handler was introduced in 2008. Since then, the company has designed and produced a wide range of turret and pick-and-place solutions for its customers all over the world.	344.0
4	Aztech Electronics Pte Ltd	SINGAPORE	TMT	consumer electronics	consumer electronics_distributor	Downstream	Being a turnkey, one-stop integrated solutions provider based in Singapore, Aztech is equipped with state-of-the-art equipment, R&D, design, manufacturing and packaging capabilities to deliver a seamless, unified experience. Each and every time. Always striving towards the edge of technology for more than 34 years, we have been building capabilities to serve clients' manufacturing needs, including the consumer electronics, telecommunications, healthtech, LED lighting, automotive and technology start-up market segments.	524.0
5	TONGDUN INTERNATIONAL PTE LTD	SINGAPORE	tmt	it_services	it_services	midstream	Tongdun Technology is a professional third-party intelligent risk management and decision-making service provider headquartered in Hangzhou, Zhejiang. By integrating artificial intelligence into business scenarios, Tongdun Technology offers solutions in intelligent user analysis, intelligent risk management, intelligent antifraud and intelligent operation to clients from various industries including financial industry, internet business, logistics, healthcare, retail, smart cities and government bodies. Over 10,000 corporate clients have chosen Tongdun Technology's products and services	593.0

The word length distribution is then plotted. As shown here, most descriptions fall between 0 to 500 characters, with the dataset being positively skewed.

```python
# now plot a distribution plot to see the word length distribution
sns.distplot(df_valid["length"], kde=True)
plt.title('Word Length Distribution')
plt.show()
```



Analysis Summary: We can see that there is a high record of the company description having a total word length of around 200 to 300, while those aboce 800 is very rare.

The NaN values remaining in the labels of the dataset are replaced with a space, so that the Model can still be trained.

4.4. Populating Nan cells

We will now have to populate Nan cells with space so that we can carry on and process with text tokenization.

```
# fill na with space instead of others
df_valid.fillna(" " ,inplace=True)
df_valid
```

	Company	Country	Sector	Subsector	Archetype	Valuechain	Company Profile Information	length
0	AKSORN CHAROEN TAT ACT. CO.,LTD.	THAILAND	TMT	media	media_aggregator/distributor	Midstream	For over 80 years of experience in creating and developing high-quality learning materials has enabled us to provide world-class educational innovation to meet the needs of all teachers, students, institutions and educational authorities.	238
1	DONGGUAN SHENGYA CLEANING APPLIAME CO.LTD	CHINA	TMT	consumer electronics	consumer electronics_distributor	Downstream	Yatai's main products cover various cleaning machinery, cleaning agents, cleaning tools, stone maintenance and other cleaning products; cleaning solutions and services include product technical consultation, product customization, employee training, maintenance and so on.	273
2	EXIS TECH SDN. BHD.	MALAYSIA	oos	others	others		In the beginning, it started off by providing technical support for test handlers, then moving into module design and production. Its first, in-house designed, full-fledged handler was introduced in 2008. Since then, the company has designed and produced a wide range of turret and pick-and-place solutions for its customers all over the world.	344
4	Aztech Electronics Pte Ltd	SINGAPORE	TMT	consumer electronics	consumer electronics_distributor	Downstream	Being a turnkey, one-stop integrated solutions provider based in Singapore, Aztech is equipped with state-of-the-art equipment, R&D, design, manufacturing and packaging capabilities to deliver a seamless, unified experience. Each and every time. Always striving towards the edge of technology for more than 34 years, we have been building capabilities to serve clients' manufacturing needs, including the consumer electronics, telecommunications, healthtech, LED lighting, automotive and technology start-up market segments.	524

Here, a number will be assigned to each unique label of the 4 targets. The numbers will then be used to train the model as the "correct answers".

4.5. Assigning tags

In this section, we will be assigning tags to every row, so that we can make use of the given keywords for bag-of-words (BoW) processing.

```
# Programmatically assign tags to each definition
sector_keywords = pd.read_excel('./sector_master_definition.xlsx')
df_keywords = sector_keywords[['Sector', 'Subsector', 'Archetype', 'Value Chain', 'Sector Keywords']]

# capitalise all tags
df_keywords['Value Chain'] = df_keywords['Value Chain'].str.upper()
df_keywords.fillna(' ', inplace=True)
df_keywords['Sector Keywords'] = df_keywords['Sector Keywords'].str.upper()
df_keywords['Sector Keywords'].replace(' ', '[]', inplace=True)

# save unique tags, sorted for consistency across runs
sector = np.sort(df_keywords['Sector'].unique())
subsector = np.sort(df_keywords['Subsector'].unique())
archetype = np.sort(df_keywords['Archetype'].unique())
valuechain = np.sort(df_keywords['Value Chain'].unique())
print(len(sector), len(subsector), len(archetype), len(valuechain))
tag_counts = [len(sector), len(subsector), len(archetype), len(valuechain)]

# assign number tag list to each row
taglist = []
for index, row in df_keywords.iterrows():
    temp = []

    temp.append(np.where(sector == row['Sector'])[0][0])
    temp.append(np.where(subsector == row['Subsector'])[0][0])
    temp.append(np.where(archetype == row['Archetype'])[0][0])
    temp.append(np.where(valuechain == row['Value Chain'])[0][0])

    taglist.append(temp)

# assign completed taglist to column in dataframe
df_keywords['list_tag'] = taglist
```

The datatypes of each row are set to "str", to prevent any conflict when training the model.

```
# we will have to ensure all the dtype of the respective columns are in string and not float for spacy to handle properly, so now we will attem
columns_to_convert = ['Sector', 'Subsector', 'Archetype', 'Valuechain', 'Company Profile Information']

for i in columns_to_convert:
  df_valid[i] = df_valid[i].astype(str)
```

The spaCy library setup occurs here. All the required libraries are imported, and a custom-built lemmatizer is added to the spaCy pipeline. The tokenisation process is also modified to not treat hyphens as punctuation, so that hyphenated words remain unsplit.

```python
# import required libraries
from spacy.language import Language
from spacy.tokens import Doc
from spacy.lang.char_classes import ALPHA, ALPHA_LOWER, ALPHA_UPPER, CONCAT_QUOTES, LIST_ELLIPSES, LIST_ICONS
from spacy.util import compile_infix_regex

# initialise nlp engine
nlp = spacy.load("en_core_web_lg")

# declare custom properties
Doc.set_extension('processed', default=True, force=True)

# Modify tokenizer infix patterns
infixes = (
    LIST_ELLIPSES
    + LIST_ICONS
    + [
        r"(?<=[0-9])[+\-\*^](?=[0-9-])",
        r"(?<=[{al}{q}])\.(?=[{au}{q}])".format(
            al=ALPHA_LOWER, au=ALPHA_UPPER, q=CONCAT_QUOTES
        ),
        r"(?<=[{a}]),(?=[{a}])".format(a=ALPHA),
        r"(?<=[{a}0-9])[:<>=/](?=[{a}])".format(a=ALPHA),
    ]
)

infix_re = compile_infix_regex(infixes)
nlp.tokenizer.infix_finditer = infix_re.finditer

# custom lemmatizer
@Language.component("custom_preprocess")
def custom_preprocess(doc):
    temp = []

    # filter through each token and add to preprocessed text if requirements #
    # met.                                                                   #
    for t in doc:
        if (not t.is_punct and not t.like_num and not t.is_stop and not t.is_digit and not (t.ent_type == 396 or t.ent_type == 397)):
            temp.append(t.lemma_.upper())

    doc._.processed = temp

    return doc

# add custom pipeline components to default pipeline
nlp.add_pipe('custom_preprocess', last=True)
```

Here, the tokenised words are added back into the dataframe for easy access.

```
[ ] # run the pipeline on data
    processed_doc = list(nlp.pipe(df_valid['Company Profile Information']))
```

```
# add lemmatised words to dataframe
df_valid['processed'] = [doc._.processed for doc in processed_doc]

df_valid
```

	Company	Country	Sector	Subsector	Archetype	Valuechain	Company Profile Information	length	list_tag
0	AKSORN CHAROEN TAT ACT. CO.,LTD.	THAILAND	TMT	media	media_aggregator/distributor	MIDSTREAM	For over 80 years of experience in creating and developing high-quality learning materials has enabled us to provide world-class educational innovation to meet the needs of all teachers, students, institutions and educational authorities.	238	[6, 18, 49, 5]

Here, the provided sector keywords are merged into a single large master list of keywords, then sorted and filtered to only include unique keywords. This master list is then used to perform the Bag-of-Words vectorisation technique to the tokenised descriptions, and the BoW vectors are subsequently added back to the dataframe.

```python
# combine all keywords from all sectors
keywords_masterlist = []
for index, row in df_keywords.iterrows():
    keywords_masterlist += eval(row['Sector Keywords'])

# remove extraneous keywords, then sort
keywords_masterlist = sorted(list(set(keywords_masterlist)))
print(len(keywords_masterlist))
```

```
1473
```

```python
# do bag of words
bow_vectors = []
for index, row in df_valid.iterrows():
    company = row['processed']

    dictionary = dict.fromkeys(keywords_masterlist, 0)
    for word in company:
        if word in keywords_masterlist:
            dictionary[word] += 1

    # append to dataframe
    bow_vectors.append(list(dictionary.values()))

    # print(f'{sum(dictionary.values()):>3}/{len(dictionary.values()):<3} |', dictionary.values())

df_valid['BoW_vectors'] = bow_vectors

df_valid
```

Example shown here:

BoW_vectors
[0, ...]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

Now, we move on to model construction. Here, we do a quick check of the tensorflow keras module before splitting up the dataset into training and testing subsets. The split ratio used here is 80/20 for train/test. The subsets were then further split into inputs and outputs.

5.1. Training Models

```
import keras

print('--- Version Checking ---')
print("Keras:", keras.__version__)
```

```
--- Version Checking ---
Keras: 2.4.3
```

```
# split datasets to train and test
distribution = int(df_valid.shape[0] * 0.8)

df_train = df_valid.iloc[:distribution]
df_test = df_valid.iloc[distribution:]

df_train.fillna(0, inplace=True)
df_test.fillna(0, inplace=True)

X_train = np.array(list(df_train['BoW_vectors']))
y_train = np.array(list(df_train['list_tag']))

X_test = np.array(list(df_test['BoW_vectors']))
y_test = np.array(list(df_test['list_tag']))
```

Here, the various functions to create the models are declared. The `multi_branch` function is to create the output layers for each label, the `create_multilabel` function is to put together the model itself, and the `one_hot` function is a custom one-hot encoder for processing the targets.

```
# create multi-output model
from keras.layers import Dense, Input, Dropout
from keras import Model

# function to build model branches
def multi_branch(x, name, input_dim, output_dim, dropout_rate):
    x = Dense(input_dim // 2, activation='relu')(x)
    x = Dropout(dropout_rate)(x)

    x = Dense(input_dim // 4, activation='relu')(x)
    x = Dropout(dropout_rate)(x)

    x = Dense(input_dim // 8, activation='relu')(x)
    x = Dropout(dropout_rate)(x)

    # output
    output = Dense(output_dim, name=name, activation='softmax')(x)

    return output

def create_multilabel(labels, labels_output_dim, input_dim, dropout_rate=0.2):
    # check labels
    assert len(labels) == len(labels_output_dim)

    input_layer = Input(input_dim)

    # group 1 dense layers
    group_1 = Dense(input_dim, activation='relu')(input_layer)
    group_1 = Dense(input_dim // 1.5, activation='relu')(group_1)

    # multilabel branches
    branches = []
    for i in range(len(labels)):
        branches.append(multi_branch(group_1, labels[i], input_dim, labels_output_dim[i], dropout_rate))

    # put model together
    model = Model(inputs=input_layer, outputs=branches, name='company_classification_model')

    return model

# one hot
def one_hot(arr, n_cat):
    output = []
    for n in arr:
        result = np.zeros(n_cat)
        result[n] = 1

        output.append(result)

    return np.array(output, dtype=int)
```

The model is created here, with the outputs named for ease of visualisation. The created model is shown in figure 27.

```
label_names = ['sector', 'subsector', 'archetype', 'valuechain']

model = create_multilabel(label_names, tag_counts, len(keywords_masterlist))

keras.utils.plot_model(model, show_shapes=True)
```

Here, the targets are one-hot encoded, and the loss metrics are defined for all model outputs.

```
# preprocess labels before training
y_train_multi = {label_names[0] : one_hot(y_train[:,0], tag_counts[0]),
                 label_names[1] : one_hot(y_train[:,1], tag_counts[1]),
                 label_names[2] : one_hot(y_train[:,2], tag_counts[2]),
                 label_names[3] : one_hot(y_train[:,3], tag_counts[3]),
                 }
y_test_multi = {label_names[0] : one_hot(y_test[:,0], tag_counts[0]),
                label_names[1] : one_hot(y_test[:,1], tag_counts[1]),
                label_names[2] : one_hot(y_test[:,2], tag_counts[2]),
                label_names[3] : one_hot(y_test[:,3], tag_counts[3]),
                }

losses = {i : 'categorical_crossentropy' for i in label_names}
```

Here, the model is trained for 200 epochs with a batch size of 20. The defined optimiser is adam, with accuracy as our evaluation metric.

```
model.compile(optimizer='adam', loss=losses, metrics=['accuracy'])
history = model.fit(X_train, y_train_multi, epochs=200, batch_size=20)
```

After training, the model is evaluated here, and their testing metrics are printed, shown below.

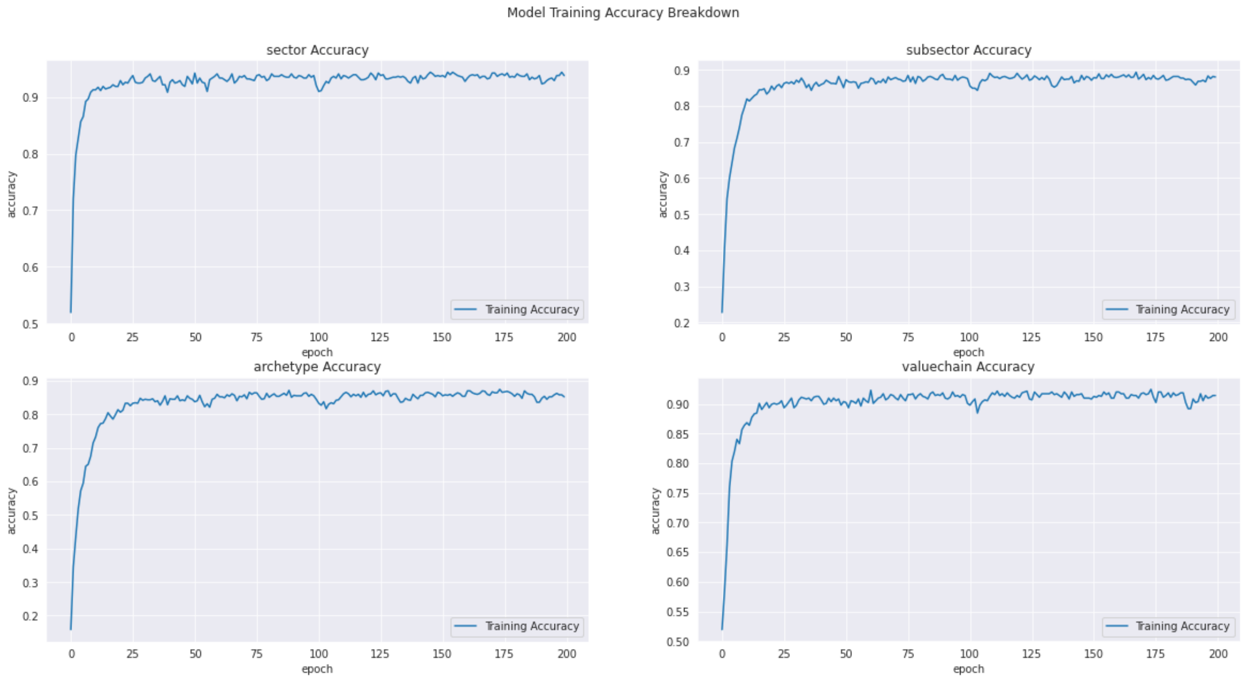
```
metrics = model.evaluate(X_test, y_test_multi, verbose=0)[1:]

for i, label in enumerate(label_names):
    print(f'{label} accuracy: {metrics[i+4] * 100:.5}%')
    print(f'{label} loss: {metrics[i]:.5}')
```

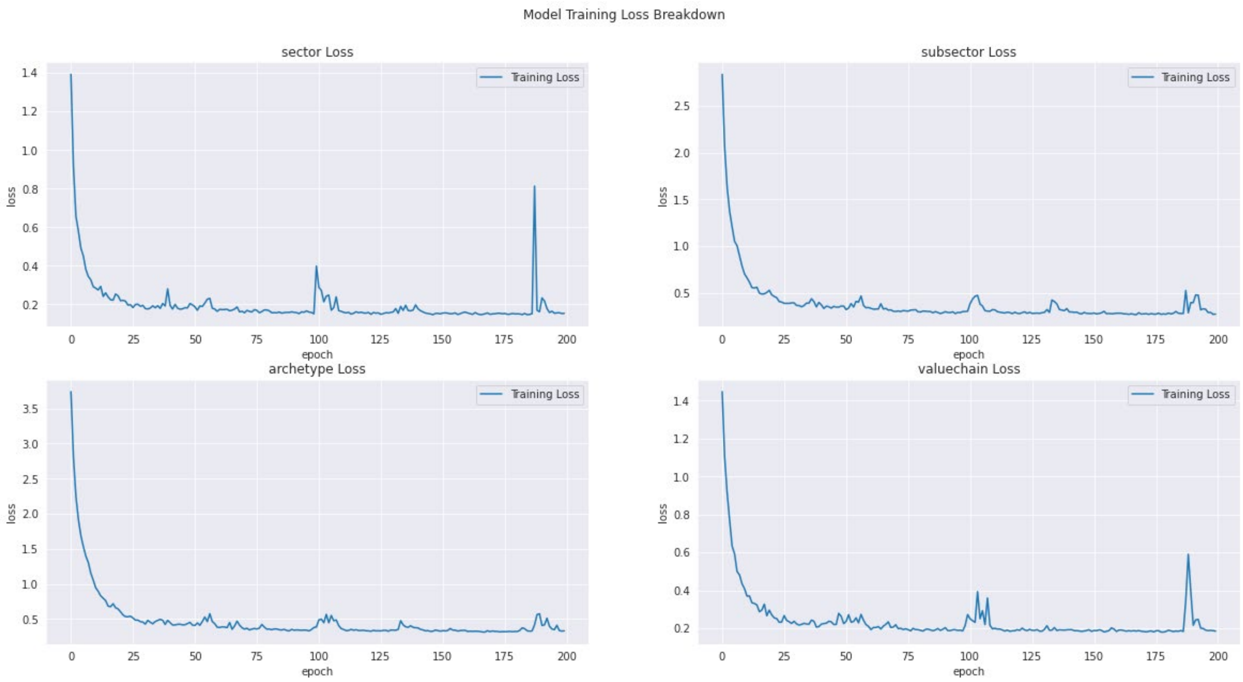
```
sector accuracy: 76.316%
sector loss: 4.6163
subsector accuracy: 59.211%
subsector loss: 4.728
archetype accuracy: 43.421%
archetype loss: 7.7406
valuechain accuracy: 60.526%
valuechain loss: 6.0506
```

Here, the accuracy and loss history of the model training is plotted.

```
[ ]
# plot accuracy and loss graphs for all labels
fig = plt.figure(1, figsize=(20,10))
plt.suptitle('Model Training Accuracy Breakdown', y=0.95)

for i, name in enumerate(label_names):
    plt.subplot(2, 2, i+1)
    plt.plot(history.history[f'{name}_accuracy'])
    plt.title(f'{name} Accuracy')
    plt.ylabel('accuracy')
    plt.xlabel('epoch')
    plt.legend(['Training Accuracy'])

fig = plt.figure(2, figsize=(20,10))
plt.suptitle('Model Training Loss Breakdown', y=0.95)

for i, name in enumerate(label_names):
    plt.subplot(2, 2, i+1)
    plt.plot(history.history[f'{name}_loss'])
    plt.title(f'{name} Loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['Training Loss'])

plt.show()
```

Shown here are the accuracy and loss history graphs.


Model Training Accuracy Breakdown
sector Accuracy
subsector Accuracy
archetype Accuracy
valuechain Accuracy
accuracy
epoch
Training Accuracy


Model Training Loss Breakdown
sector Loss
subsector Loss
archetype Loss
valuechain Loss
loss
epoch
Training Loss

Here, a test is done with the testing labels to see the predicted results of the model.

```
[ ]
results = model.predict(X_test)

predicted_label = []
for label in results:
    predicted_label.append(np.argmax(label, axis = 1))
predicted_label = np.array(predicted_label)

for i in range(predicted_label.shape[1]):
    print(f'Expected: {y_test[i]} | got {predicted_label[:,i]}')
```

Shown below is a partial output.

```
Expected: [ 6 29 26  7] | got [ 6 29 25  5]
Expected: [ 4 24 64  0] | got [ 3 25 69  3]
Expected: [ 1 31 89  7] | got [ 1 31 40  3]
Expected: [ 4 24 64  0] | got [ 1  4 14  5]
Expected: [ 4 24 64  0] | got [1 3 8 5]
Expected: [ 1  4 10  5] | got [ 1  4 10  5]
Expected: [ 6 17 46  5] | got [ 6 17 46  5]
Expected: [ 3 21 63  3] | got [ 3 21 63  3]
Expected: [ 1  6 14  5] | got [ 1  4 10  5]
Expected: [ 1  4 10  5] | got [ 1  4 10  5]
Expected: [ 1  4 10  5] | got [ 1  4 10  5]
Expected: [ 5 26 75  7] | got [ 1  6 14  5]
Expected: [ 6  8 20  3] | got [ 6  8 20  3]
Expected: [ 3 16 45  1] | got [ 3 21 63  3]
Expected: [ 4 24 64  0] | got [ 6  7 15  3]
Expected: [ 1  4 10  5] | got [ 1  4 10  5]
Expected: [ 3 21 63  3] | got [ 3 21 63  3]
Expected: [2 2 5 6] | got [ 4 24 64  0]
Expected: [ 1  4 10  5] | got [ 1  4 10  5]
```