# National College of Ireland
# Project Submission Sheet
# School of Computing

National College of Ireland

| | |
|---|---|
| **Student Name:** | Dawn Walsh, Komal Riddhish Bharadva, Marcelo Fischer, Parth Adesh Darekar, Prasad Rudrappa Shivu, Sachin Harishchandra Nikam |
| **Student ID:** | x19190352, x19213051, x20118872, x19212739, x19213077, x19198159 |
| **Programme:** | Data Analytics |
| **Year:** | 2021 |
| **Module:** | Data Mining and Machine Learning II |
| **Supervisor:** | Anu Sahni & Michael Bradford |
| **Submission Due Date:** | 13/05/2021 |
| **Project Title:** | Meta-tag Generation for SEO using LSTM models |
| **Page Count:** | 16 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | Dawn Walsh, Marcelo Fischer, Komal Riddhish Baradva, Parth Adesh Darekar, Prasad Rudrappa Shivu, Sachin Harishchandra Nikam |
| **Date:** | 20th May 2021 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | □ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Meta-tag Generation for SEO using LSTM models

Dawn Walsh
*School of Computing*
*National College of Ireland*
Dublin, Ireland
x19190352@student.ncirl.ie

Komal Riddhish Bharadva
*School of Computing*
*National College of Ireland*
Dublin, Ireland
x19213051@student.ncirl.ie

Marcelo Fischer
*School of Computing*
*National College of Ireland*
Dublin, Ireland
x20118872@student.ncirl.ie

Parth Adesh Darekar
*School of Computing*
*National College of Ireland*
Dublin, Ireland
x19212739@student.ncirl.ie

Prasad Rudrappa Shivu
*School of Computing*
*National College of Ireland*
Dublin, Ireland
x19213077@student.ncirl.ie

Sachin Harishchandra Nikam
*School of Computing*
*National College of Ireland*
Dublin, Ireland
x19198159@student.ncirl.ie

***Abstract*—People increasingly rely on search engines to find relevant information as the World Wide Web grows in size. It is the responsibility of the search engine to provide the user with relevant and high-quality information in response to their query. The main goal of this research report is to aid a Fintech Jobs portal attract as many companies as possible to their website, by enhancing the visibility of their portal with the help of meta-tags specifically created for job search criteria.The goal was to examine if the site rankings as well as the hits of the site are increased. For this research work we have implemented a recurrent neural network model with the help of a supervised text generation technique. This research offers a comprehensive overview and step-by-step implementation of the selected search engine optimization strategies that have been shown to improve visibility, attract more users, and achieve higher rankings in search results for a Fintech job portal website with the help of different meta-tags generated from the model.**

***Index Terms*—Meta-tags, SEO, LSTM, Neural network, Text-generation, Fintech jobs**

## I. Introduction

The 21st century is known as the era of information. Technology is advancing at an astonishing pace, faster than ever before in the history of man. Unlike in the past, currently most businesses make use of these freely available and cheap technologies to widen their reach and attract more customers by using the internet and various social media platforms. Some businesses operate only online and the company itself is a social media page or a website. However, the easy access to all of those technologies enables a huge number of people to attempt to do the same and it becomes easy to get relegated to just background noise.

The focus of this report is a job board company which specialises in a specific employment area. Job board companies generate revenue from advertising job vacancies for different companies, they most often operate exclusively online, utilising various social media platforms, such as LinkedIn, or via their own website. This kind of service is popular with job hunters because they facilitate the process of finding jobs that are scattered all over the internet and make it easier to focus and filter the vacancies by type, region, city, country, among others. Some well known job boards are:

1) LinkedIn[1] which is not just a job board, it is also a social media platform, but with a higher emphasis on professional connections and networking opportunities,
2) Glassdoor[2] which not only advertises jobs but also allows current and former employees to post feedback about the type of company it is, the culture and the pros and cons of working there.
3) Indeed[3] which is more of a traditional job board.

This research will focus on Fintech Jobs[4] which is a specialist job board company focused on, perhaps unsurprisingly, the Fintech arena. They advertise vacancies for multiple companies and have their job offers powered by Jobbio[5]. The business is completely online and must therefore overcome the hurdle of competing with thousands of other similar and oftentimes much bigger websites which have teams of people whose sole job is selling advertising space on their website. For these types of businesses, being on top of search engines such as Google and Bing is critically important in order to successfully attract more customers and users. The issue that the team was tasked with was helping the company to improve its online presence in order to attract more companies that would advertise on their website.

On the surface this is a Marketing problem and we are a team of Data Analytics students however in order

[1] www.linkedin.com/
[2] www.glassdoor.ie/Reviews/index.htm
[3] https://ie.indeed.com/?r=us
[4] https://fintechjobs.io/
[5] https://jobbio.com/

to perform effective marketing first one must understand the market better which is hopefully where we might prove useful by providing insights. After some initial experimentation it was noted that the ranking of the company in search engines is not optimal. The use of different search terms might find the website in rank 10 or sometimes the website does not even appear in the first page. It is worth noting that more than 90% of people who use search engines do not go past the first page of results. In light of the above, this work will focus on answering the following research question:

- **RQ:** Can meta-tags generated by a recurrent neural network (RNN) using a supervised text generation technique based on the top ranking websites produce specific meta-tags and improve the rank of Fintech Jobs in search engines such as Google?

The rest of this report is structured as follows: in section II we discuss other works done in the field and highlight strengths and weaknesses. In section III we provide an overview of the methodology used for this project. Section IV discusses the details of the data gathering phase, the implemented technique and the deployment phase. In section V we highlight our results and evaluation metrics and discuss them in detail. Finally, in section VI we summarize our findings and propose some guidance in further improving upon this project.

## II. Related Work

In the 1990s, search engines were not as successful as they are now since they were mostly based on keyword matching and back links. As a result it was relatively simple for low-quality websites to rank higher by focusing on their exact keywords and building a large number of back links. To address this problem, Google developed an algorithm to filter the results and clean the site. Since then, Google's algorithm has been updated on a regular basis in order to preserve and increase the performance of its search engine. The most basic component for search engine value is title marks, which are the most basic component for search engine rankings. The title tag is found in the HTML record's head section, and it is the most important piece of ”meta” data about a page in terms of importance and positioning [1]. The Meta description tag on a web page provides short, concise information about the page's material. It appears after the title in search engine results pages on a regular basis (SERP). The Meta description tag is important because you can use it to convey your marketing message and encourage search engine users to click on your posting rather than the competition's [2].

Improving web pages' rankings on Google or Bing is considered the holy-grail for most businesses. Some companies are content to pay for the privilege, which is why most of Google's revenue comes not from innovation but from advertising sales. However, more companies would rather optimize their website so that it stays on that all important front page of the search results. This has resulted in numerous studies being carried out on the best methods for doing this and has its own skill set in the form of Search Engine Optimization (SEO). It is known that ranking factors and how they impact web page ranking are a topic of intense research, and while search engines do publish some of their ranking factors and best practice for using them they do not publish them all or indeed how the factors are weighted within the ranking formula.

SEO is still almost as much of an art as it is a science as those that specialise in this area need to have a good understanding of both web page building and search engine algorithms. There has been some research into using machine learning methods on improving the aspects of a website that an SEO will generally concern themselves with [3] [4]. These factors tend to be what is called organic or on-page factors such as:

1) Text and information quality
2) Ease of Navigation
3) URL and domain name
4) Meta-tags
5) Short image descriptions
6) Heading information and so on

All of these things are often tweaked and changed manually and may take several days or even months to actually bear fruit in the search engine ranking war. In [3] the authors concerned themselves with classifying web pages into low, medium or high quality clusters, they initially did this manually be employing experts to rank the sites chosen based off the published SEO best-practice guidelines. This information was then used alongside several different data-mining techniques to try to extract the factors that were most relevant to produce a high-ranking page. Their results indicated that meta-tags were among the most important factors alongside title and H1 (heading). Their research suggest up to 3 meta-tags as being the optimum number.

Further research by [5] into SEO shows work on the Google search algorithm named ”HUMMINGBIRD”. ”Hummingbird” prioritizes natural language queries over individual keywords, emphasizing context and significance. It also digs deeper into content on individual websites, giving it a better chance of directing visitors to the most relevant pages rather than just the homepage. Structural workflows then followed to identify meta-tags from the initial step of understanding a business, following optimization to search engines and extracting keywords using the name of the most important files to improve the content of tags. A small case study is proposed to understand the appearance of pages and factors affecting the display of page ranks to extract values at different levels of intervals. To initiate the methodology, researchers used web crawler names Google Spider algorithm which is clueless about language text and page size and setting evaluation with multiple parameters using statistics methods and tests.

A study by [6] aims to suggest the use of metadata for achieving a high ranking in the Search Engine Result Page (SERP) by incorporating SEO best practice. It also aims at an approach to recommend metadata which follows these two steps of combining metadata and keywords from high-ranking websites and accessing the significance of terms based on semantic relevance. The main study focuses on on-page optimization technique as a proposed method to optimize websites. It was concluded that Hill top algorithm which is famous page rank optimizer algorithm shows best results with the working of queries. It has received a growing amount of traffic when metadata and keywords are combined. As the work focuses on text data like HTML files, further research was concluded to increase the work in non-text data to crawl images and video data as well.

Currently, Recurrent Neural Networks (RNNs) are a popular method for creating both translation models and text generation models [7]. They can be difficult to both understand and explain, however they use previous inputs alongside current inputs to determine their outputs. See [8] for a comprehensive and in-depth look at the structure of RNNs.

Long Short Term Memory (LSTM) networks are a version of RNN that is currently the subject of much research. The use of interdependent inputs means that they are very useful at processing sequential data to improve understanding of natural language (NL). However when it comes to generating text, relevancy and context are key. We can see this from [9] which looks at using RNN and LSTM to generate new stories from input stories. This research relied on a huge body of work being provided for the inputs to the network. Each input story consisted of approximately 5000 words. The output stories were evaluated by human readers to see if they could pass the test of being good enough, it was shown that those that did not pass had problems with grammar and context.

In a separate study by [10] the authors were unconcerned with the performance of the model, they were interested in the semantic closeness of the output to the input rather than the quality of the output text. As is often the case when it comes to text generating networks over-fitting was a serious problem, achieving high training accuracy while the validation accuracy never really gets off the ground.

It is important to note that in some business areas, such as tourism, companies have found website optimization to be an ineffective tool as giants such as Expedia will just pay to have the entire front page of hits on Google or Bing, if they decide to create a focused advertising campaign in a specific region/country. In smaller and more specialised areas such as job advertising there is still a lot to be gained from optimising a website and utilising whatever insights can be gained in as many ways as a company possibly can.

## III. Methodology

The data mining methodology used for this project was the Cross Industry Standard Process for Data Mining (CRISP-DM). It is a process that consists of hierarchical steps, going from general to specific [11]. The basic CRISP-DM flow structure depicts the whole life cycle of a data mining project and can be seen in Fig. 1.

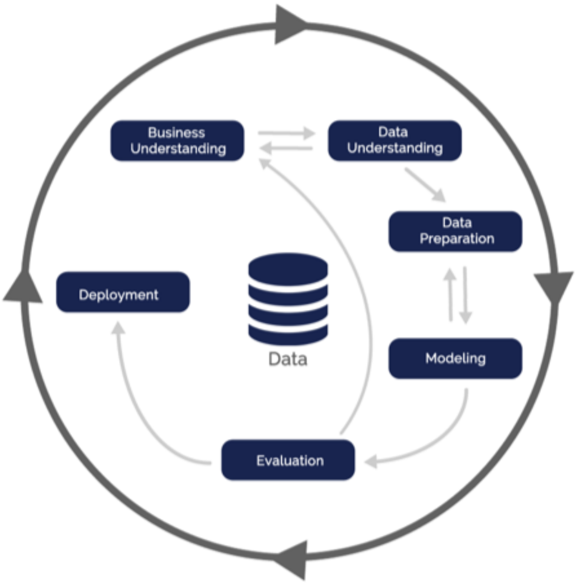


Figure 1. Life cycle of the CRISP-DM methodology. Image was taken from https://smartvision-me.com/wp-content/uploads/2019/08/crisp-dm.png [accessed on 05/05/2021]

Each step is summarized as follows:

- **Business Understanding:** This is one of the crucial steps. It consists of understanding what the problem is in terms of the business perspective. Then it needs to be translated into a data mining problem, specifying the preliminary project plans.
- **Data Understanding:** This step begins by gathering data related to the business problem. Followed by gaining familiarity with the data, identify patterns, inconsistencies, redundancies, and get some initial insights.
- **Data Preparation:** This is the most important step in order to get a satisfactory and working final model. A commonly heard phrase within the data mining field is ”garbage in, garbage out”. At this stage all the steps that have been taken are done with the aim of building the final dataset which will be fed into the machine learning model. All the hard data cleaning, feature engineering, feature extracting, and transformations are carried out here. It is important to note that the tasks mentioned above will be executed more than once before this phase is complete.

- **Modeling:** This step consists of selecting a range of different machine learning or statistical techniques and applying them to the output data of the previous step. Improving the models via parameter tuning or by going back to the data preparation phase is also part of this step. Note that there is a strong connection between the data preparation and the modeling phases.
- **Evaluation:** After applying different models to the final dataset it is crucial that the models have a reasonably high quality from a data perspective. But even so, it is needed to go back to previous steps and guarantee that none of the businesses rules were broken while searching for the optimal model and that all businesses requirements were considered. Once the review is done, the use of the results of the modeling phase can be discussed.
- **Deployment:** Upon the successful creation of a model it is extremely important to present it to the customer in a way that they can understand it and use it. The deployment might be a simple report or a fully functioning application with several capabilities. Therefore, it is crucial to have in mind prior to building the models what they will be used for.

For this project, the business understanding phase was done by having meetings with the client and agreeing upon a specific project objective. It was agreed that the deployment phase would consist of a system that could take search terms as inputs and generate meta-tags based on the terms given. This way the client is able to choose a set of search terms that they would like to optimize their rank for. The remaining CRISP-DM steps will be explained in the next section.

## IV. Implementation

This section will give details on how the project was implemented and all of the considerations and assumptions that were made.

### *A. Data Gathering*

Quite often, getting the data needed is the most challenging part of a data mining project. In our case, the company did not have any prior dataset built and no useful data available for the project proposed. Thus, we brainstormed different ideas and ended up choosing to web scrape all that we would need. Since the objective was to build a model that could take search terms as inputs to generate meta-tags as outputs, it was crucial to have a set of search terms, together with the resulting websites' ranks and their respective meta-tags. The Python libraries used for this first part were:

- nltk (natural language toolkit)
- Beautiful Soup
- re (regular expressions)
- pickle (to save Python objects)
- wikipedia

The client's company is focused on the fintech market, therefore we chose to web scrape two web pages related to the topic: the Investopedia page for fintech[6] and the Wikipedia page for fintech[7]. After extracting all the text from both pages, several cleaning steps were carried out such as removing any characters that were not letters or numbers, converting all letters to lower case, removing words with digits, removing digits, removing inverted commas, and removing unicode characters. Then, we tokenized (separated the text into a list of single words) the resulting clean text and removed any stopwords given that they add no meaning. After this, we built our first glossary comprised of all the words that remained from both web pages and merged them together in a set to remove any duplicates. However, we noticed that a lot of the words were very similar, e.g. 'acting', 'active', 'actively', 'activities', 'activity'. Hence, we decided to use lemmatisation (reduce every word to its lemma) to try and remove similar words that would not bring new results when used as search terms. With this, we made a new set of words with the lemmatised version and this was our final glossary of terms to be used in the search engines.

Once we had constructed the glossary we started generating several lists containing bi-grams and tri-grams (two- and three-word phrases) comprised of words taken from the glossary. Our first try was to randomly pick words from the glossary to generate the lists, but we soon realized that from all of the bi/tri-grams in the lists only a few were directly related to fintech. Our second approach was to force some specific terms inside each list, therefore giving different weights to the words (two examples for the tri-grams are: "<random word from glossary>fintech recruitment" or "fintech dublin <random word from glossary>"). We generated a total of 2 bi-grams lists, each with 500 elements, and 6 tri-grams lists, each with size 400. Then, we used the Python library `googlesearch` to send queries to Google using all the lists and their respective terms and we saved the first 10 links from each search, preserving their ranking order. At the end of this step, we had eight new lists containing all of the top 10 links for each search term, one for each of the bi-grams' lists and one for each the tri-grams' lists.

The next thing we did was to collect the meta-tags from all of the links that we gathered on the previous step. Here we used the following Python libraries:

- Numpy
- Beautiful Soup
- Requests
- concurrent.futures (enables threading or multiprocessing)

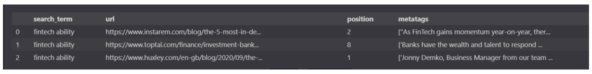
[6] www.investopedia.com/terms/f/fintech.asp

[7] https://en.wikipedia.org/wiki/Financial_technology

- pickle (to save Python objects)

The initial attempt involved looping through each of the lists containing the links and extracting two specific meta-tags, namely description and keywords. The first is related to the description of the web page and its overall content, while the later is related to specific keywords that the owners of the web page consider to be important to the topic of the website. We decided to consider only these two types of meta-tags for our approach since they are the most relevant when describing the web page. We used the `requests` library to access each link in the list and then extract the set of meta-tags discussed above. Initially, we looped over the links' list and accessed one by one, sequentially. However, this implementation took more than 14 hours to run for some of the lists. We then implemented a new version of the code using Threading. This made possible for multiple queries concurrently and we were able to reduce the time needed for the script to run by almost 70%.

The final step for the data gathering part was to put together all of the pieces of information we got from the web into a master dataframe. The final dataframe is composed of 4 columns: search_term, url, position (is the rank) and metatags. The first 3 rows of the dataframe can be seen in Fig. 2.



| | search_term | url | position | metatags |
|---|---|---|---|---|
| 0 | fintech ability | https://www.instarem.com/blog/the-5-most-in-de... | 2 | ["As FinTech gains momentum year-on-year, ther... |
| 1 | fintech ability | https://www.toptal.com/finance/investment-bank... | 8 | ['Banks have the wealth and talent to respond ... |
| 2 | fintech ability | https://www.huxley.com/en-gb/blog/2020/09/the-... | 1 | ['Jonny Demko, Business Manager from our team ... |

Figure 2. First 3 rows of the final dataframe.

### B. Data Preparation

Prior to implementing any model it is crucial to clean and prepare the raw data into a suitable format without garbage inside. Since our data is comprised of text, we used very similar steps to the ones we used before for the glossary. As can be seen in Fig. 2, the meta-tags were saved as Python lists. We firstly removed the square brackets and the quotation marks from both ends. If the website did not have the meta-tags we were looking for, the program would return an empty list. On encountering an empty list it was replaced with NaN (not a number) values. Any symbols such as $ were removed at this stage. When trying to access the web pages it was also possible to encounter blocked links or the page took too long to access. In the case of blocked links the value "Link blocked" was saved, in the case of timing out the value "Connection timed out" was saved. Since both do not have a set of meta-tags, they would not be useful for the objectives of the project and were simply dropped. The final step prior to modeling was to drop any duplicates in the url column, since they would certainly have the same set of meta-tags.

### C. Modeling

After data gathering and data preparation, the modeling for the final dataset was carried out. Initially the modeling was carried out on the complete dataframe without any filtering being performed on it. The punctuation and ASCII characters were removed, if present, and a cleaned corpus was created. Then a tokenizer was used to apply the tokenization on the cleaned corpus and convert the data to a sequence of tokens. The input data given to an LSTM model for training and validation should be of the same length and dimension. One of the studies shows that using pre-padding for the LSTM network is more efficient [12]. With this in mind it was decided to use pre-padding to the maximise the length of the input data. Since the size of the input data is very large, a sample of the input data was taken to feed into the model. We have divided the data into train and test parts in the 90:10 ratio. Finally, a sequential LSTM model was built which includes the input Embedding layer, the Hidden layer which is an LSTM layer, and an output Dense layer.

The input layer which is an embedding layer produces dense vectors of fixed size from a set of positive integers given. Many studies have used LSTM models for word or character-based inputs [13] [14] [15]. These studies show the use of the Embedding layer proves to be very useful for prediction in language modeling. The output vector of the Embedding layer is a dense vector having real values in place of 0's and 1's. The benefits of this are that these values then represent the words in a better sense with dimensionality reduction [16]. The required parameters to the embedding layer include input dimension, output dimension, and input length. The input dimension is the vocabulary size that is 8796 in our case. The output dimension is the vector space size in which the words will be embedded. We have set the output dimension as 100 for the size of the vector space. Lastly, the input length is the length of sequences of input which is 222 for our model.

The hidden layer is next which contains the LSTM or the bidirectional LSTM layer or a combination of both. The input to an LSTM model should be a 3D tensor that has batch size, time-steps, and features. Each LSTM layer should go along with a dropout layer which helps in preventing overfitting. Ideally, the dropout should be 20% which means it ignores 20% of the neurons while training. In this project, we have used different combinations of LSTM and bidirectional-LSTM layers in an attempt to achieve a good validation accuracy and this will be discussed further in the results section.

The output layer is the Dense layer. The output of the dense layer will be equal to the number of neurons specified in this layer. We have set this to the total words which is the vocabulary size that is 8796. The output of this layer is a weighted linear combination of the input with bias. This layer is used for outputting a prediction. This is also the point where the activation function of the model is defined. The activation function is responsible for activating the neuron within the network by calculating its weighted sum and adding bias. It presents non-linearity in the neuron output. In different models both softmax and sigmoid were considered as activation functions in an effort to find as strong a model as possible.

The next step is to compile the model. The loss, optimizer, and metrics which are the hyper-parameters, are specified. The loss function is the error function that is used to calculate the loss of the model to update the weights so that loss can be minimized on the next evaluation. Categorical cross-entropy was chosen as the loss function in most of the combinations of models applied. Categorical cross entropy is used for multi-class classification applications and was considered to be the most appropriate loss function for the model. Moreover, the optimizer chosen in this study is the Adam optimizer as it is generally considered to be the best choice in these types of problems. The metric chosen was accuracy as it is suitable for multi-class classification.

Finally, the model was fit with X_train, Y_train, epochs, batch_size, and validation_split. The epoch is the count for which the learning algorithm will process through the whole training dataset. Each epoch consists of batches. The number of samples progressed before the model gets updated is batch size and the number of full passes through the entire training dataset is the number of epochs. We have used different epochs and batch sizes for various grouping of models. The LSTM model by Keras helps in dividing the training data into validation sets so that we can assess the model performance on the validation data. The hyper-parameter validation_split can be set to 20% or 30% for validation purposes. Various arrangements of 10%, 20%, and 30% validation set sizes were considered on multiple models.

The model was evaluated using X_test and Y_test dataset and the accuracy is calculated for the same. Usually, training a deep learning neural networks takes a large amount of time especially if the capacity of the hardware does not meet the specific requirements, which most personal machines do not. Once the model has been built and trained, it is saved to a file. This saved model can be used later to make the required predictions which will eventually save a lot of time. Keras provides 3 different formats in which the model can be saved namely:

1) JSON format
2) YAML format
3) HDF5 format.

The JSON and YAML formats store only the model structure while the HDF5 format stores the complete model not only its structure but its weights as well. Therefore, we have used HDF5 format to store our model(s). We have also used Netron to visualise the model. Netron is a viewer for deep learning neural networks and machine learning models, it works with several different file formats including Tensorflow and Keras. Figure 3 shows the summary of different layers included in the built model.

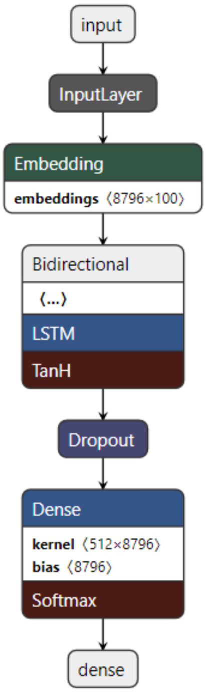
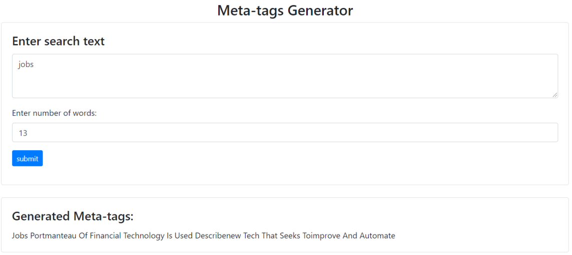


Figure 3. Summary of the built model.

### D. Deployment

After Data modeling, the final step is deployment. The deployment is an application of the model which is capable of taking inputs and providing relevant outputs. This is very useful in making business decisions and helps the end-user or customer to make use of the built model for prediction. There are multiple ways in which one can deploy the model in the production environment. The saved model has been used to predict the outcome displayed on the graphical user interface (GUI). For deployment purposes, the flask framework was used which provides a development server for hosting the application. Figure 4 shows the predicted output on the GUI.


Meta-tags Generator

Enter search text

jobs

Enter number of words:

13

Generated Meta-tags:

Jobs Portmanteau Of Financial Technology Is Used Describenew Tech That Seeks Toimprove And Automate

Figure 4. GUI output.

## V. Discussion

### A. Evaluation and Results

The effectiveness of various RNN architectures in terms of text generation is discussed in this section. On Google's open source data flow engine Colab using TensorFlow, all experiments were developed using Long Short Term Memory (LSTM), RNN's with "ADAM" optimizer. TensorFlow helps programmers to run computations on a variety of platforms, including several CPUs, GPUs, and mobile devices. All experiments are run on GPU-enabled TensorFlow to speed up the computations. Various configurations of experiments were carried out to find the optimal values for parameters such as learning rate, number of units / memory blocks, and network structure such as number of hidden recurrent layers. We started with a medium-sized LSTM network, which has an input layer, a hidden recurrent layer, and an output layer. Multiple combinations were used to run the network with different network units from 0-1024, loss functions as binary cross entropy and categorical cross entropy, Adam and RMSProp as an optimizer, softmax and sigmoid as an loss function. All experiments were executed for between 100 and 300 Epochs depending on how long they were taking to run and whether or not they were showing promise. Some were stopped early if it was clear that they were not improving.

The following network topologies are used to pick the most suitable RNN/LSTM/GRU network structure for training models.

- LSTM/Bi-directional LSTM 1 layer with 0 - 1024 units
- LSTM/Bi-directional LSTM 2 layers with 0 - 1024 units
- LSTM/Bi-directional LSTM 3 layers with 0 - 1024 units

Experiments were run for up to 1024 units for each of the network topologies. Each network topology used different epochs to learn the patterns that lead to the generation of important meta-tags. It was observed that when there were less than 25 epochs remaining, the simple LSTM and 3 Layered LSTM networks began to overfit. This was a strong indicator that the network had begun to memorize the training data samples, resulting in a decrease in the generating samples generalization efficiency.

The implementation of recurrent neural networks and extensive study of the model yielded a reasonable result. However, there are significant improvements to be made. Accuracy is used as the evaluation metric to assess the model which is given by.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where,

1) TP - True Positive
2) TN - True Negative
3) FP - False Positive
4) FN - False Negative

In total 26 different models were tried, a selection of them are discussed below.

*1) Results for Single Layer models Bi-LSTM:* The initial model created had top 10 meta_ tags which involved 3 layers all being Bi-LSTM. Layers 1 and 2 having 256, and layer 3 having 128 units set. The number of units is a parameter in the LSTM that refers to the hidden state's dimensionality and the output state's dimensionality (they must be equal). Since the weight matrix allows cross-talk between the hidden states, it is not accurate to think of it as the serial LSTMs operating in parallel. Softmax activation function is used in the initial model. The Softmax activation function is used as a neural networks last activation function to normalize the networks output to a probability distribution over the expected output group. Categorical cross entropy (CCE) was used as a loss function with 1 epoch. As was expected it did not yield a good result, train accuracy was about 4.6% whereas the validation accuracy was about 4.4%. The validation split and test-train split were 0.1 each. Removing the second and third layers from the model, started yielding better results.

- Single layer Bi-LSTM with 100 units and 0.1 dropout used Softmax function at 100 epochs and Categorical Cross Entropy loss function showed a little over 97% train accuracy and 51% validation accuracy. The validation split and test-train split were 0.2 each.
- Single layer Bi-LSTM with 100 units and 0.2 dropout used Sigmoid function at 100 epochs and Categorical Cross Entropy loss function showed about 97% train accuracy and about 55% validation accuracy. Validation split and test-train split were reset to 0.1 each.
- When units were set to 128 and dropout was increased to 0.25, and epochs decreased to 50, the model, using Softmax function and CCE loss function, yielding about 81% train accuracy and over 49% train accuracy, when validation split, and test-train split were 0.2 and 0.1, respectively.

Retaining the above combinations, the model was run at 100 epochs. The accuracy increased to 93% and 54% on validation and test-train respectively.

*2) Results for LSTM Models:* The next iterations were carried out using single layer LSTM model. With 100 units, 0.1 dropout, and softmax activation function, the model was run for 100 epochs with the BCE loss function. The results showed over 92% train accuracy and over 56% validation accuracy. When the units were increased to 512 and dropout to 0.2, the model showed almost 98.5% train accuracy and 55% validation accuracy, using CCE loss function

*3) Results for Multi-Layer models:* Different combinations were tried on two- and three-layer RNN models.

- Three layer model – 2 layers of LSTM and 1 layer of GRU, at 512, 256 and 128 units, 0.3, 0.25 and 0.1 dropouts, run at 150 epochs using CCE loss function and Sigmoid activation function yielded 97% and 44% accurate train and test results respectively.
- Two-layer LSTM with 256 and 100 units and 0.25 dropout, run at 150 epochs using Softmax activation function and CCE loss function showed 80% train accuracy and 54% validation accuracy.

Changing the units to 512 and 256, and loss function and activation function to BCE and Sigmoid, the train and test accuracies increased to 98% and 60% respectively. This is the best result obtained of all combinations.

The graph in Fig. 5 shows the training and testing accuracy for the model. These results show that the accuracy obtained on the test set are extremely good, however the validation set accuracy indicates that the model is overfitting.

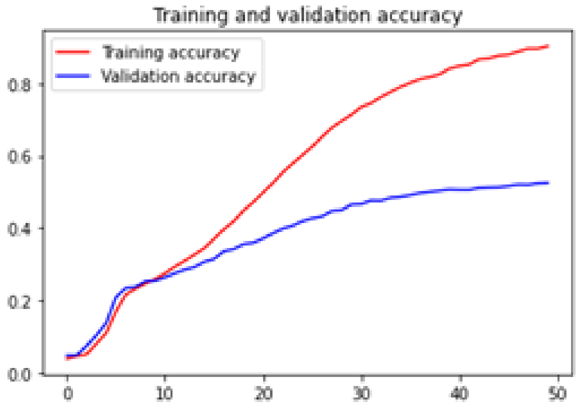


Figure 5. Training and Validation Accuracy

The graph in Fig. 6 shows the training and testing loss for the data. The values obtained show that the loss is comparatively very low for the training set, however the validation set loss starts high and does not improve over time, again indicating that the model is overfitting.

The computation time of this and most of the other models was extremely high, especially without the use of the GPU in Colab with some of the more complex models taking over 14 hours to run outside of a GPU, it can be concluded that the model is an alright fit for the data however it is clear that there are improvements to be made.

The implementation of multiple hidden layers did not increase the accuracy or the efficiency of the trained data, it can be seen in Fig. 8 that the best performing models had fewer layers but higher number of initial input units. There was also a considerable improvement over other models on its performance on the validation set data. The output produced was also sensible at least on the surface Fig. 7 it is readable and references Fintech.

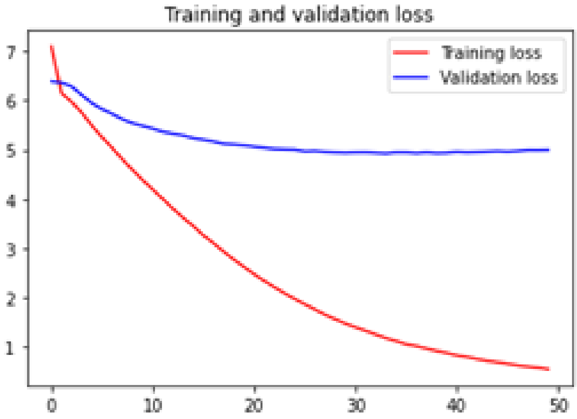
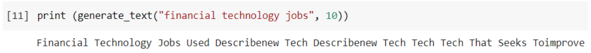


Figure 6. Training and Validation Loss



```
[11] print (generate_text("financial technology jobs", 10))

     Financial Technology Jobs Used Describenew Tech Describenew Tech Tech Tech That Seeks Toimprove
```

Figure 7. Model Output

However it is evident that significant improvements could be made with a larger dataset and perhaps hyperparameter tuning and optimisation.

## VI. Conclusions and Future Work

Many-to-many RNNs are extremely greedy, not only do they require a high volume of data, that data also needs to be high quality. So while the data that was generated and used was of good quality there simply was not have enough of it. In order to improve the outcomes of this research a much larger dataset would be required and many more search-terms would need to be created. Perhaps an order of magnitude or more larger. This would require possibly weeks of web-scraping. The other major stumbling block with running an RNN is the amount of processing power that they require, and the larger the dataset is the more they need. If access to a GPU could be guaranteed many larger and more complex models could be attempted. There are also significant gains to be made by tuning the hyperparameters of the model, however this is a time consuming process and is often as much an art as a science.

Besides that, after the data was gathered and merged it into a master dataframe there was no distinction made between description and keyword meta-tags, they were just concatenated together. If it was possible to get more data, two different models could be created: one specifically to generate keywords and another one to generate description meta-tags. This could greatly improve the generated texts and make them more context specific.

Another point to be considered is that the model generates words based on previous words. Word based models usually require huge amounts of data, much more than character based models. So whilst the dataset used was of a considerable size, due to technological constraints, only a small proportion of the dataset could be considered, otherwise Google Colab would crash and kick us out. Therefore the accuracy is lower due to data size. Character based modelling can also be considered which provides correct sequences grammatically, but it comes with its own problems like it requires a bigger hidden layer and is computationally more expensive.

## References

[1] S. Eric, "The art of seo (illustrated ed.)," 2009.

[2] J. Kristopher, "Search engine optimization: Your visual blueprint for effective," 2010.

[3] G. Matošević, J. Dobša, and D. Mladenić, "Using machine learning for web page classification in search engine optimization," *Future Internet*, vol. 13, no. 1, p. 9, 2021.

[4] J. Salminen, J. Corporan, R. Marttila, T. Salenius, and B. J. Jansen, "Using machine learning to predict ranking of webpages in the gift industry: Factors for search-engine optimization," (New York, NY, USA), Association for Computing Machinery, 2019.

[5] A. Kakkar, R. Majumdar, and A. Kumar, "Search engine optimization: A game of page ranking," in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 206–210, 2015.

[6] S. An and J. J. Jung, "A heuristic approach on metadata recommendation for search engine optimization," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, p. e5407, 2021.

[7] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," *Journal of King Saud University - Computer and Information Sciences*, 2020.

[8] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, Mar 2020.

[9] D. Pawade, A. M. Sakhapara, M. Jain, N. Jain, and K. Gada, "Story scrambler - automatic text generation using word level rnn-lstm," *International Journal of Information Technology and Computer Science*, vol. 10, pp. 44–53, 2018.

[10] S. Santhanam, "Context based text-generation using lstm networks," 2020.

[11] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1, Springer-Verlag London, UK, 2000.

[12] M. Dwarampudi and N. Reddy, "Effects of padding on lstms and cnns," *arXiv preprint arXiv:1903.07288*, 2019.

[13] M. Cho, J. Ha, C. Park, and S. Park, "Combinatorial feature embedding based on cnn and lstm for biomedical named entity recognition," *Journal of biomedical informatics*, vol. 103, p. 103381, 2020.

[14] C. Lu, H. Huang, P. Jian, D. Wang, and Y.-D. Guo, "A p-lstm neural network for sentiment classification," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 524–533, Springer, 2017.

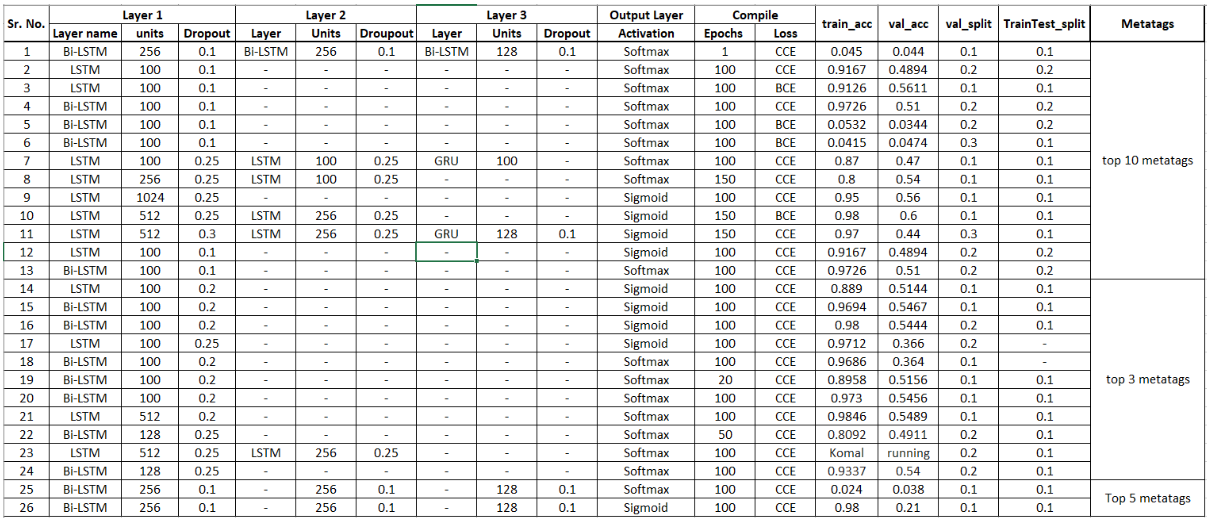
[15] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pp. 747–754, 2017.

[16] S. Saxena, "Understanding embedding layer in keras," 2020.

## VII. Appendix

### A. Model Summary

Figure 8 shows the various combinations of the model tries with their corresponding training and validation accuracy. The Adam optimizer was constant throughout the experiment. There were some more models tried but got kicked off from google Colab in the middle. So, those models are not included in this model summary.



| Sr. No. | Layer 1 | | | Layer 2 | | | Layer 3 | | | Output Layer | Compile | | train_acc | val_acc | val_split | TrainTest_split | Metatags |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Layer name | units | Dropout | Layer | Units | Droupout | Layer | Units | Dropout | Activation | Epochs | Loss | | | | | |
| 1 | Bi-LSTM | 256 | 0.1 | Bi-LSTM | 256 | 0.1 | Bi-LSTM | 128 | 0.1 | Softmax | 1 | CCE | 0.045 | 0.044 | 0.1 | 0.1 | top 10 metatags |
| 2 | LSTM | 100 | 0.1 | - | - | - | - | - | - | Softmax | 100 | CCE | 0.9167 | 0.4894 | 0.2 | 0.2 | |
| 3 | LSTM | 100 | 0.1 | - | - | - | - | - | - | Softmax | 100 | BCE | 0.9126 | 0.5611 | 0.1 | 0.1 | |
| 4 | Bi-LSTM | 100 | 0.1 | - | - | - | - | - | - | Softmax | 100 | CCE | 0.9726 | 0.51 | 0.2 | 0.2 | |
| 5 | Bi-LSTM | 100 | 0.1 | - | - | - | - | - | - | Softmax | 100 | BCE | 0.0532 | 0.0344 | 0.2 | 0.2 | |
| 6 | Bi-LSTM | 100 | 0.1 | - | - | - | - | - | - | Softmax | 100 | BCE | 0.0415 | 0.0474 | 0.3 | 0.1 | |
| 7 | LSTM | 100 | 0.25 | LSTM | 100 | 0.25 | GRU | 100 | - | Softmax | 100 | CCE | 0.87 | 0.47 | 0.1 | 0.1 | |
| 8 | LSTM | 256 | 0.25 | LSTM | 100 | 0.25 | - | - | - | Softmax | 150 | CCE | 0.8 | 0.54 | 0.1 | 0.1 | |
| 9 | LSTM | 1024 | 0.25 | - | - | - | - | - | - | Sigmoid | 100 | CCE | 0.95 | 0.56 | 0.1 | 0.1 | |
| 10 | LSTM | 512 | 0.25 | LSTM | 256 | 0.25 | - | - | - | Sigmoid | 150 | BCE | 0.98 | 0.6 | 0.1 | 0.1 | |
| 11 | LSTM | 512 | 0.3 | LSTM | 256 | 0.25 | GRU | 128 | 0.1 | Sigmoid | 150 | CCE | 0.97 | 0.44 | 0.3 | 0.1 | |
| 12 | LSTM | 100 | 0.1 | - | - | - | - | - | - | Sigmoid | 100 | CCE | 0.9167 | 0.4894 | 0.2 | 0.2 | |
| 13 | Bi-LSTM | 100 | 0.1 | - | - | - | - | - | - | Softmax | 100 | CCE | 0.9726 | 0.51 | 0.2 | 0.2 | |
| 14 | LSTM | 100 | 0.2 | - | - | - | - | - | - | Sigmoid | 100 | CCE | 0.889 | 0.5144 | 0.1 | 0.1 | top 3 metatags |
| 15 | Bi-LSTM | 100 | 0.2 | - | - | - | - | - | - | Sigmoid | 100 | CCE | 0.9694 | 0.5467 | 0.1 | 0.1 | |
| 16 | Bi-LSTM | 100 | 0.2 | - | - | - | - | - | - | Sigmoid | 100 | CCE | 0.98 | 0.5444 | 0.2 | 0.1 | |
| 17 | LSTM | 100 | 0.25 | - | - | - | - | - | - | Sigmoid | 100 | CCE | 0.9712 | 0.366 | 0.2 | - | |
| 18 | Bi-LSTM | 100 | 0.2 | - | - | - | - | - | - | Softmax | 100 | CCE | 0.9686 | 0.364 | 0.1 | - | |
| 19 | Bi-LSTM | 100 | 0.2 | - | - | - | - | - | - | Softmax | 20 | CCE | 0.8958 | 0.5156 | 0.1 | 0.1 | |
| 20 | Bi-LSTM | 100 | 0.2 | - | - | - | - | - | - | Softmax | 100 | CCE | 0.973 | 0.5456 | 0.1 | 0.1 | |
| 21 | LSTM | 512 | 0.2 | - | - | - | - | - | - | Softmax | 100 | CCE | 0.9846 | 0.5489 | 0.1 | 0.1 | |
| 22 | Bi-LSTM | 128 | 0.25 | - | - | - | - | - | - | Softmax | 50 | CCE | 0.8092 | 0.4911 | 0.2 | 0.1 | |
| 23 | LSTM | 512 | 0.25 | LSTM | 256 | 0.25 | - | - | - | Softmax | 100 | CCE | Komal | running | 0.2 | 0.1 | |
| 24 | Bi-LSTM | 128 | 0.25 | - | - | - | - | - | - | Softmax | 100 | CCE | 0.9337 | 0.54 | 0.2 | 0.1 | |
| 25 | Bi-LSTM | 256 | 0.1 | - | 256 | 0.1 | - | 128 | 0.1 | Softmax | 100 | CCE | 0.024 | 0.038 | 0.1 | 0.1 | Top 5 metatags |
| 26 | Bi-LSTM | 256 | 0.1 | - | 256 | 0.1 | - | 128 | 0.1 | Sigmoid | 100 | CCE | 0.98 | 0.21 | 0.1 | 0.1 | |

Figure 8. Model Summary

Assignment Details:

Produce a plan / roadmap for your client enterprise.

Word Count: 5000 (Excluding tables, images, references and appendices)

Produce a presentation summarising your plan / roadmap

Approximately 5-10 slides

Optional Extras:

Individual reflective journal

Peer to peer marking

The purpose of the plan / roadmap is to address the challenge / opportunity as defined by and agreed with the client enterprise. You are required to work as a group to formulate a tangible plan which sets out what the client enterprise should do, why and how. In order to do this, you will identify challenges, opportunities, options and issues; research, evaluate and ultimately recommend option(s) for how the enterprise should proceed.

Depending on your course, module and the client enterprise your plan may focus on marketing, innovation, design thinking, new product / service development, business development, opportunity identification and evaluation, commercialisation, and / or new venture creation. The plan should include opportunity identification and evaluation, market research and strategy development. You may wish to include financial projections and funding requirements (where appropriate).

Suggested content headings for the plan are provided below. Please note that not all need to be included and the order can be arranged as required. (Sections marked with a * are optional and should be included where they are deemed relevant and where the client enterprise is willing to provide background information).

Suggested Plan Content:

- Executive Summary
- Context
  - Company (History / Structure / Product / Services)
  - Challenge / Opportunity Description
  - Analysis (Company / Market / Competitor / Industry / Micro / Macro / Consumer)
  - Research & Evaluation
- Strategy
  - Strategy (Marketing / Sales / Commercialisation / Innovation etc.)
    - Recommendations / Options / Objectives / Actions
    - Funding Requirements
    - Staffing & Operations *
    - Financial Projections *
    - Metrics / Implementation Controls / Key Performance Indicators
    - Strategy Timeline/ Milestones
- References
- Appendices

# ENFUSE

## Group Consultancy Project Brief

Lecturer: Dr Anu Sahni & Dr Michael Bradford

Module: Data Mining & Machine Learning II

Assignment Type: Group Consultancy Project

Assignment Issue Date:

Assignment Deadline:

Assignment Weighting:

ENFUSE Overview:

ENFUSE matches enterprises and social enterprises in Dublin with teams of Masters Level - University Students. During semester 2 (January-April) and as part of a course module, students work in teams of approximately 4-6 people with selected enterprises to help provide insights, propose solutions and ultimately present a bespoke and tangible plan that sets out how challenges and opportunities could be addressed by the enterprise. The plan and student work is aligned to a core module(s) of the students' course and is submitted for academic grading as well as to the enterprise. A pitch competition is held at the end of ENFUSE where shortlisted and finalist student teams represent their class / university and compete by pitching in front of industry / expert judges on how their assigned enterprise could address challenges and realise opportunities.

ENFUSE Aims:

- Support micro / small-medium enterprises by providing bespoke plans / roadmaps to address challenges and develop opportunities
- Provide participating students with real-world consultancy and enterprise experience
- Develop enterprise ecosystem links and synergies between stakeholders and enterprises

Student Learning Outcomes:

On successful completion of this assignment, you will be able to:

- Perform a strategic assessment of a live enterprise scenario
- Conduct detailed research using secondary and primary data sources
- Analyse and interpret data and formulate tangible recommendations
- Produce and present on a research based plan
- Work collaboratively and effectively as part of a team

Please note, the below confidentiality agreement must be included in the plan and signed by each student team member and client enterprise.

Confidentiality Agreement

The undersigned students acknowledge that the information provided to them to facilitate the creation of this plan is confidential; therefore, the undersigned students agree not to disclose it without the express written permission of the undersigned client enterprise.

It is further acknowledged by the students that information to be furnished in this plan is in all respects confidential in nature, other than information that is in the public domain through other means and that any disclosure or use of this confidential information by the reader may cause serious harm or damage to the client enterprise; therefore, the students agree not to disclose it without the express written permission of the client enterprise. This is a suggested plan only and does not imply offering of securities.

It is further agreed by the client enterprise that the plan resulting from this assignment may be disclosed to relevant staff in Technological University Dublin / Dublin City University / National College of Ireland and Dublin City Council in order to enable grading of the student's work as part of their course and as part of ENFUSE.

Student Group

| Student Name | Signature | Date |
|---|---|---|
| Dawn Walsh | | 13/05/2021 |
| Marcelo Fischer | Marcelo Fischer | 13/05/2021 |
| Komal Riddhish Bharadva | | 13/05/2021 |
| Sachin Harishchandra Nikam | | 13/05/2021 |
| Parth Adesh Darekar | parthdarekar | 13/05/2021 |
| Prasad Rudrappa Shivu | | 13/05/2021 |

Fintechjobs.io

| Company Signatory | Signature | Date |
|---|---|---|
| DAVID CROWLEY | David Crowley | 15/05/202 |

Strictly private and confidential. This document is the proprietary property of Fintechjobs.io.

Copying or otherwise distributing the information contained herein is a breach of confidentiality agreement.

Project Steps:

1. Client Enterprise and Team Allocation

Each student team will be allocated a client enterprise / can select from the list of applicant enterprises that they wish to work with. The team will be provided with initial briefing material on their client enterprise including: enterprise overview, challenge / opportunity overview and contact details.

- Please nominate a member of your team who will be the lead communicator with the client enterprise.
- Contact the client enterprise as soon as possible and send a short portfolio featuring a bio of each team member (skills / interests / experience), overview of your course / module and suggestions on how your team can assist.
- Before your first meeting, review the client enterprises' overviews and conduct some initial research / idea generation / brainstorming

2. Agree Project Brief

At the first client enterprise meeting, the student team and enterprise will formulate a project brief. The brief can include: project background / overview, objectives, outputs required, methodology, estimate of timings etc. Please ensure to formulate a brief that incorporates meaningful and realistic outputs.

- Please inform and agree the project brief with your lecturer before issuing to the client enterprise.

3. Client Enterprise Meetings

Each student team should meet at least four times with their client enterprise during the assignment timeline. Regular meetings provide a vital way to ensure that effective communication take place between the enterprise and student team. These meetings provide a formal means to review progress, clarify details and check that the work completed is in line with the expectations of the client enterprise. Meetings can be face-to-face* or virtual and should be arranged directly between the enterprise and the student team. It is the responsibility of the student team to arrange meetings.

*Dependent on government restrictions and university health and safety advice, meetings may take place physically. (Please review up to date government restrictions regarding meeting physically before arranging etc.) A meeting should take place during each of the project phases noted below.

Please ensure that the following tasks are completed regarding all meetings

- Agree date / time  with client enterprise (Nominated lead communicator should do this)
- Record minutes of each meeting
- Agree actions (To do, Doing, Done)
- Share minutes with client enterprise after each meeting

| ENFUSE 2021 – Timeline (Subject to change) | | | | | |
|---|---|---|---|---|---|
| Phase | 1 | 2 | 3 | 4 | 5 |
| Meeting | Initial briefing (Agree brief) | Progress Update | Progress Update / Present draft plan | Submit Plan to Client Enterprise & for grading | *Pitch Competition |
| Month | February | March | March | April | May (TBC) |

*A Pitch Competition will be held at the end of ENFUSE for all participant students and enterprises. Shortlisted and finalist student teams will represent their class / university. These teams will compete by pitching in front of industry / expert judges on how their assigned enterprise could address challenges and realise opportunities. Before pitching, the shortlisted student teams must inform and obtain permission from their client enterprise regarding pitch content. At the event, a winning student team will be chosen by a panel of expert judges. (Further briefing will be given to the shortlisted teams before the pitch competition).

4. Complete and Submit Plan / Roadmap

Use the guidance above to help prepare your plan / roadmap. Use appropriate methods / theories and models to analyse the context and options for the client enterprise. Please ensure that the recommendations are tangible and relevant to the enterprise.

5. Presentation

Prepare a presentation which summarises your plan / roadmap. This can be used when presenting your plan to your enterprise and for your in-class presentation. Shortlisted student teams from the module will proceed as finalists at the ENFUSE Pitch Competition.

At this competition, student teams will represent their class / university and compete by using their presentation to pitch in front of industry / expert judges on how their assigned enterprise could address challenges and realise opportunities.