Natural Language Processing for the Analysis Sentiment using a LSTM Model

Achraf BERRAJAA
Euromed Research Center, Euromed University of Fes, Morocco

***Abstract*—Over the past decade, social networks have revolutionised the communication between organisations and their customers, and the data provided by customers on social network platforms is having an increasingly important impact on how organisations collect and analyse this data to make better decisions. We have prepared a new dataset that will allow the scientific community to estimate and evaluate new models using nearly the same conditions. Moreover, this dataset represents a recent and interesting sample for the proposed machine learning models to correctly identify the topics or points on which the company should focus to improve customer satisfaction and better meet their needs. Therefore, we have proposed a recurrent neural network (RNN) with Long short-term memory (LSTM) that we will run in the cloud to predict sentiment analysis. The objective is also to define systems capable of extracting subjective information from natural language texts, such as feelings and opinions, with the aim of creating structured knowledge that can be used by a decision support system or a decision maker for better customer management. The proposed neural network has been trained on the proposed dataset which contains 50 000 customer observations. The performance of the proposed architecture is very important as the success rate is 96%.**

***Keywords*—Artificial intelligence; NLP; RNN; LSTM; customer relationship management**

I. Introduction

In recent years, social networks have revolutionised communication between organisations and their customers. The data provided by customers on social media platforms is having an increasing impact on how organisations collect and analyse this data to make better decisions. Natural language processing (NLP) is one of the most promising ways to process data and text from social networks. Developing powerful methods and models to extract relevant information from large amounts of data from multiple sources and languages is a complex challenge. It can be overcome when a powerful pipeline is built to transform this raw data into useful information that we can use.

Information extraction [27], classification and grouping methods [25] are one of the main approaches to leverage raw textual data and transform it into valuable information. In the field of data processing, a robust pipeline is needed to clean this textual data and make it ready for use by different models. To this end, our objective is to propose a data preparation pipeline ranging from data processing and cleaning to digital representation of textual data. The prepared data is intended to be used as training data for machine learning models. A deep learning model is also implemented for sentiment analysis, defined as a system for extracting subjective information from natural language texts, such as feelings and opinions, to create structured knowledge that will be used by a decision support system or a decision maker.

As far as structure is concerned, this paper is organised as follows: Section 2, presents a literature review on the works related to the different technologies used for natural language processing and their fields of use, namely linguistics and artificial intelligence. Section 3 details the structure of our data preparation pipeline which consists of six modules: data cleaning, tokenization, data normalization, stemming and lemmatization, token categorization and finally data representation. Section 4 is reserved for the creation of machine learning models that are able to classify the observations in our data by assigning them a star score ranging from one to five (Very Satisfied, Satisfied, Fair, Dissatisfied and Very Dissatisfied). In Section 5, a Recurrent Neural Network (RNN) with Long Short-Term Memory cells (LSTM) is implemented for the sentiment analysis of our customers. Section 6 is devoted to the digital experiments. We also discuss and comment on the results obtained and show the effectiveness of our models. Finally, a conclusion that summarises the study and gives some potential perspectives for the developed approach to improve the current results.

II. Related Work

Social networking is a phenomenon that has recently developed worldwide and has rapidly attracted billions of users. The main reason for this phenomenon is the ability of online social networks to provide a platform for users for better communication, as explained in the work of [16]. This form of electronic communication through social networking platforms allows customers to generate their content and share it in different forms, usually in text form. This content is very valuable to the organisations involved. Therefore, automatic natural language processing is formed as an emerging area of research and development [19].

Natural Language Processing can be defined as a field of study using computer science, artificial intelligence and linguistic concepts to analyse natural language. In other words, NLP is a set of tools used to derive meaningful information from textual data and generally used to obtain knowledge and decision support by processing textual data present in web pages, documents, customer reviews [20].

As mentioned earlier, linguistics is an essential part of natural language processing and can be defined as the scientific study of the structure and development of language with particular emphasis on grammar, semantics and phonetics. In other words, linguistics is primarily concerned with the design and evaluation of language rules. If we read this definition carefully, we realise that natural language is controlled by a set of rules such as grammar and semantics. These rules will be a key factor in enabling the machine to understand the textual data and to process it. The author in [13] present a study that summarises the linguistic research techniques used in automating the analysis of the linguistic structure of language and, at the same time, sheds light on the development of core technologies such as speech recognition, speech synthesis and machine translation using artificial intelligence.

Artificial intelligence (AI) is a branch of computer science that aims to propose and construct systems able of performing tasks that require human intelligence. In other words, any algorithm or computer technique capable of performing sophisticated tasks such as driving a car or diagnosing a disease can be classified as artificial intelligence (for more information, see [24]). Machine learning is a sub-field of artificial intelligence that deals with the development of algorithms capable of learning to perform tasks automatically on the basis of a large number of examples, without being pre-programmed to do so in the case of supervised learning, or with the creation of clusters and grouping of the most similar observations in the case of unsupervised learning. We cite the book *Machine learning algorithms* [8] for more information on machine learning algorithms. On the other hand, Deep Learning refers to the branch of machine learning based on neural network architectures. In [17], the authors summarise the basic principles of deep learning and machine learning to provide a general understanding of the methodical foundations of current intelligent systems. The development of NLP applications relies heavily on machine learning and deep learning methods. Therefore, both disciplines play an important role in the development of this project.

In a natural language processing project, the data is nothing more than text - unstructured data produced by people to be understood by others. Nevertheless, this unstructured data contains patterns or indications that allow it to be analysed by a computer. To put it differently, although text is unstructured data, it is not disordered. On the contrary, text is governed by linguistic properties that enable communication [10]. Thus, our contribution will be to find patterns in the text, and analyse them for better decision making. This approach is similar to that of [20] who proposed a framework for big data analytics in commercial social networks for sentiment analysis and fake review detection for marketing decision making. Furthermore, when working with natural language, we frequently encounter the concepts of syntax and semantics. The syntax of a language refers to the rules that govern the way in which linguistic elements are put together to form phrases, clauses, and sentences. It should be noted, however, that the syntactic rules of a natural language are not as strict as those of computer languages. This is means that it is essential that a sentence follows the basic syntactic rules, so that it can be used to enable the machine to process textual data.

Natural language processing is performed on text data ranging from a few words entered by the user for an Internet search to multiple documents to be analysed and from which information needs to be extracted for better decision making. Thus, natural language processing is used in a variety of situations to solve many different types of problems:

- Searching for a string of characters: automatic processing of natural language makes it possible to identify specific elements in the text. For example, it can be used to find the occurrence of a word or more generally a string of characters in a document. As an illustration, in the work of [21], a string recognition method based on a lexical search method was proposed. In this method, the string is identified by searching a sequence of segment patterns matching the string in a lexicon.

- Entity recognition: this involves extracting names of places, people, organisations and products from the text. As an application, [26] proposed an NLP pipeline from part-of-speech tagging, through chunking, to named entity recognition of Twitter.

- Sentiment analysis: This technique is used to determine people's feelings and attitudes towards a product or service. It is useful for providing feedback on how a product or service has been perceived. Companies in all sectors are analysing their social network data streams to better understand their customers' opinions. The main challenge is to extract reliable textual reviews from consumers and use them automatically to evaluate the best products or brands. [20] proposed a framework to automatically analyse reviews. Sentiment analysis was used to analyse online reviews on Amazon. Similarly, in the work of Kumar and Sebastian (2012), a hybrid method is proposed, which uses corpus-based and dictionary-based algorithms to define the semantic orientation of opinion words in tweets. In [3], a new method of approaching sentiment classification is proposed based on Twitter data.

- Search engine: Making extensive use of natural language processing for a variety of tasks, such as query understanding, query expansion, question answering, information retrieval, ranking and clustering of results. For example, in [1], a search engine specific to scientific research is proposed, which first collects the information disseminated on the web in the sites of academic institutions and in the personal homepages of researchers. Then, after intensive text processing, it summarises the information in an enriched and user-friendly presentation oriented towards non-expert users. A question answering system has been proposed by [23], which consists of automatically answering a question posed by a human in natural language using a pre-structured database or a collection of natural language documents.

- Electronic messaging: Email platforms use natural language processing to provide a series of features, such as spam classification, inbox priority, auto-completion. As an example for spam classification, [18] proposed a comprehensive spam classification system based on semantic text classification using NLP and URL-based filtering. Similarly, [12] compared various ML algorithms and a convolutional neural network was studied with the objective of creating a powerful and efficient model for correctly classifying emails.

The above use cases are just a sample. NLP is increasingly used in several other applications, and new applications of NLP are emerging very rapidly. For instance, program synthesis is an ambitious and new field that consists in generating a source code (programming code) from a natural description. The author in [6] proposed an RNN with LSTM cells that generates java source code from a description expressed in natural language.

Several factors make the process of processing text data difficult. The existence of several languages, each of them having different rules [10] is an example of this difficulty. To illustrate, words can be ambiguous and their meaning may depend on their context. A word can also express an action, a feeling, a name or something very different. As textual data belong to the family of unstructured data, it is necessary to go through a set of data preparation operations to make them useful and usable. Despite these difficulties, natural language processing is able to perform complicated tasks in an adequate way and to bring added value in many domains. For example, sentiment analysis can be performed on customer reviews, which can identify potential problems and anomalies with a certain product or service and improve it. In the following, a robust pipeline is used to clean up the textual data and make it ready for use by the various AI models. Postal services have been considered as an application. A pipeline is proposed for data preparation, ranging from data processing and cleaning to digital representation of textual data.

III. Dataset, Exploratory Data Analysis and Pipeline

One of the most important applications where customer service needs to be improved is the postal service. Thus, our goal is to project this application to improve this service.

A. The Dataset

The first step in the development process of any NLP classification system is to collect data relevant to the proposed problem. An initial idea that may arise is to use structured datasets found on sites such as kaggle, but the problem with this approach is that these datasets deal with topics that are not relevant to the business knowledge related to the services offered by the Post. This will result in poor performance for our model. Therefore, we will need to think about retrieving a meaningful and real training dataset for the Post. One way to build a meaningful dataset is to collect reviews from organizations that operate in very similar, if not identical, domains to the Post using a scrapping method. The idea of choosing several organizations instead of one is on the one hand to have a large volume of data and on the other hand, because very often an organization tends to have mainly positive opinions and at the same time some negative opinions (having diffuse opinions).

In order to build the most accurate and robust model possible, it is necessary to have a fairly generalized set of data and to touch as much as possible all categories of reviews. To do so, several datasets from different countries (United States Postal Service, Canada Post and French Post) have been collected. Fig. 1 shows the percentage of observations of the data with a score ranging from 1 to 5 (note that we collect and scrap the data in open source).

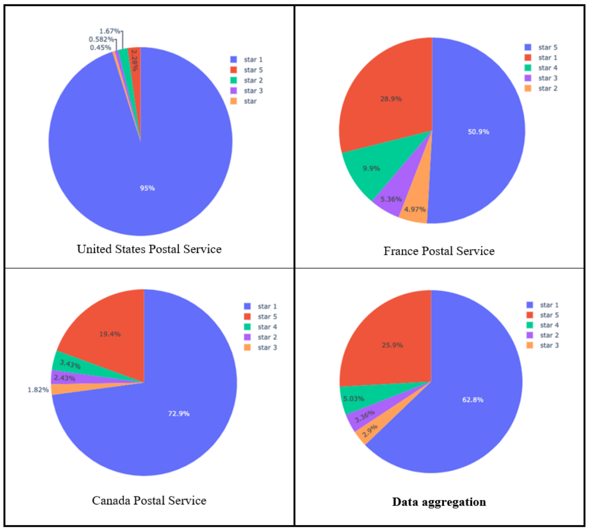

Fig. 1. The Percentage of Observations with a Score Ranging from 1 to 5.

Due to the fact that Post France notices are in French. The Google API was used to translate the data observations to obtain standardized observations in one language, English.

B. Exploratory Data Analysis

The idea behind this part (topology of text data observations) is to get an idea of how the data was constructed and if there is some pattern in the way reviews are written by customers, i.e. words used by customers, length of comments, etc. This will help us choose the Hyper-parameters for the Machine Learning / Deep Learning models. We start by visualizing the length distribution of the data observations in Fig. 2. To do this, we can calculate the length of each review and then visualize them using a histogram (Fig. 3).

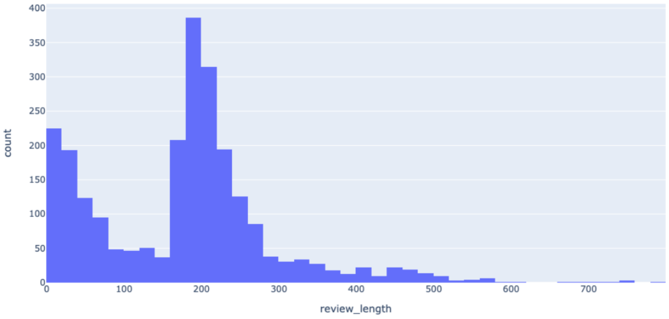

Fig. 2. Histogram of the Length of Characters per Observations.

We can say that most comments are between 180 and 219 characters long, and as the number of characters' increases, these comments become rarer. On average, a comment contains about 176 characters with a median of 190.

We can see that most of the customer reviews are about 30 to 39 words long, with an average of 32 words per review and a median of 34 and a maximum of 151 words.

This will allow us to have a first interaction with the textual data we have in order to extract indications and a first general

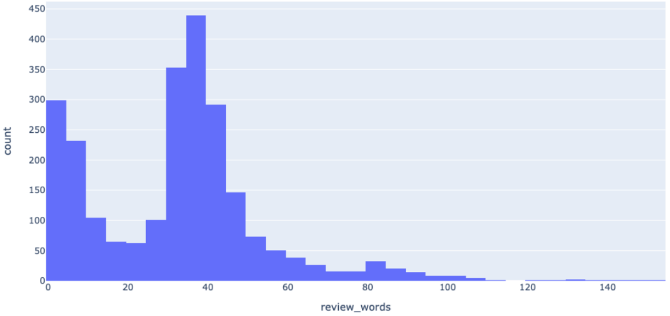

Fig. 3. Histogram Representing the Number of Words per Observation.

view of the available data.

C. Data Preparation Pipeline

At this point, the goal is to clean the data to make it more usable. This task consists of a series of steps to process, clean and normalize the textual data into a form that contains much less noise. The general idea is to remove unnecessary elements that may interfere with decision making or handicap the learning process.

The structure of our data preparation pipeline consists of six modules: data cleaning, tokenization, data normalization, stemming and lemmatization, token categorization and finally data representation.

1) Data Cleaning: Regular expressions allow us to create string patterns and use them to find or substitute specific strings in textual data. Python offers a rich module named ”re” for creating and using regular expressions [15].

The idea behind this step is to remove characters and symbols that are usually non-alphanumeric characters, which add extra noise to text data. Regular expressions are used to detect non-alphanumeric characters and remove them afterwards.

2) Tokenization: Tokenization is the process of dividing a text into ”tokens”, a token being a significant unit of text, very often a word, that we wish to use to perform an analysis. This step is essential in any natural language processing project, especially with ML models. The importance of this task will become even more apparent when representing textual data in digital format.

We performed two types of tokenization: a) Sentence tokenization is the process of breaking up textual data into sentences. A user review may or may not contain several touching sentences about the same topic, the idea is to try to structure the observation into meaningful sentences. In order to perform tokenization, we will use the *nltk* library using the *nltk.senttokenize()* function, which has been pre-trained and gives impressive results. b) Word tokenization is the process of breaking a sentence down into its component parts. A sentence is a collection of words, and with tokenization, we basically break the sentence down into a list of words.

3) Data Standardization: In order to make the NLP system more robust, it is essential to go through this step where we will normalize the data by making all the tokens lowercase or uppercase. We can see the value of this step in case we are looking for patterns in our data, or we are trying to match a certain word, but the main usefulness of this step is seen in the Feature Engineering step, where we will represent our textual data in a digital format. If this step is omitted during the data cleaning process, we will get a very high dimensional vector representation that will handicap the learning models considerably. We apply this process to all data observations using the *lower()* method.

4) Stemming and Lemmatization: Stemming and lemmatization ensure that the different forms a word can take, i.e. plurality, gender, cardinality, tense, etc., are treated as single vector components, which reduces the feature space during digital representation and makes the models more efficient during learning.

Stemming is important in the sense that it helps normalize words to their basic root, which facilitates many applications such as classification or any other ML algorithm. This task is performed via the ”*SnowballStemmer*” where we specify an input language which for our project is obviously English, the stemmer removes suffixes and endings from the words and transforms them to their basic form.

A lemma is the basic form of a word. In other words, it is the form in which the word would appear if it were listed in a dictionary. The result of stemming is not always a correct word, but the lemma of a word will always be present in the dictionary. In the application of lemmatization, an additional step is involved where the formed root is compared to the information provided by the dictionary and if and only if the lemma is present in the dictionary it can be taken into consideration. This makes undoubtedly this step much more complex and computationally demanding. The research proposed in [5] is based on comparing the accuracy performance of document retrieval based on language modeling technology, especially stemming and lemmatization. In addition, a baseline ranking algorithm was used to compare the two technologies.

D. Categorization of Tokens

1) Part-of-Speech Tagging: Part-of-speech (POS) is a process that determines the grammatical category of each token in the textual data. It labels nouns, verbs, adjectives, adverbs, interjections, conjunctions, singular nouns, plural nouns, proper nouns, etc. All this is done by models trained on datasets containing tokens and the associated tag. This is a crucial step for many NLP applications because by identifying the POS of a word, one can infer its meaning in context, which is used by many machine learning models to better classify textual data. The same token can sometimes be a noun, a verb or adverb. This is where Part-of-Speech tagging becomes of great importance since it allows us to categorize tokens as precisely as possible based on a statistical approach. To add, we use the *POS* model of the *Spacy* library to categorize the tokens of textual observations.

The same word can have different interpretations. This is why grammatical structure is important. The *NLP spaCy* library uses models that have been pre-trained to best predict the usage of a word in a data observation. When analyzing customer reviews, we want to know what actions customers have performed or undergone in order to get to know them better and this is where POS tagging comes in, allowing us to easily retrieve all actions performed by the customer.

2) Named Entity Recognition: When examining textual data observations, we tend to first identify the key actors in the observation, such as people, places, and organizations. This classification helps us break up data observations into entities and make sense of the semantics of the data observation. The Named Entity Recognition (NER) mimics the same behavior and is used to classify entities (tokens).

Named Entity Recognition (NER) is a process that detects names in textual data, as well as dates, monetary amounts and other types of entities. NER tools often focus on three types of categories: Person, Organization, and Location, drawing on models that have been pre-trained. The Named Entity Recognition process consists of two sub-tasks: entity detection (the token) and entity type determination based on statistical methods, very often supervised models. NER models rely on several parameters, one of which is the POS (part-of-speech-tagging) of the token. It is at this point that we can see a clear relationship between the POS and NER process.

E. Data Representation

Feature engineering is an important step for any machine learning problem. No matter how good the learning algorithm used, if we introduce bad features, we will get bad results.

At this point, we only have cleaned textual data, but we cannot feed it to a machine learning model. Therefore, it is essential to find a way to represent this data so that we can process it with a given model. In other words, we need to transform our textual data into digital form so that it can be passed to ML algorithms.

In this task, the objective was to work on different methods of representing text as vectors in order to choose the best one for our learning models. Several data representations have been used in the field of data science, such as Bag-of-Words [28], TF-IDF [2], Word Embeddings [4] and Word2vec [11]. We chose to use TF-IDF because of its compatibility with this problem.

TF-IDF aims to quantify the importance of a given word relative to other words for each observation in our dataset. The problem with the bag-of-words is that, since the feature vectors are based on the frequency of tokens, some terms may appear frequently in all observations in our dataset and tend to mask the importance of other words in the feature set. In particular, words that do not appear as frequently, but may be more interesting and important as characteristics. A simple way to think about the TF-IDF process is as follows: if a word m appears many times in an observation, but does not appear much in the rest of the observations in the dataset, then word m should have high importance for the observation in question. The importance of m should increase in proportion to its frequency in the observation in question, but at the same time its importance should decrease in proportion to the frequency of the word in the other observations.

Mathematically, TF-IDF is the product of two metrics. TF measures the frequency of occurrence of a word in an observation. Since different observations in the dataset may be of different lengths, a term may appear more often in a long observation than in a short observation. To normalize this, we divide the number of occurrences by the length of the observation. The IDF measures the importance of a word in the entire dataset. We calculate it by dividing the total number of observations in our dataset by the number of observations containing the term. We do this for each term and then apply a logarithmic scale to the result.

The TF-IDF score is a product of these two metrics. Thus, the TF-IDF score = TF * IDF. And can be represented as follows:

$$W_{i,j} = tf_{i,j} \times Log(\frac{N}{df_i})$$

We apply this same model to our dataset and this numerical representation will be used several times over the prediction models.

IV. Construction of a Review Score Prediction Model

At this stage, we explored our data, and were interested in the syntax, structure and semantics of the observations in our dataset. We also cleaned our data; then the data was represented in a digital format using the TF-IDF algorithm. The next step is to build models that allow us to take advantage of all the steps seen in the previous tasks. In this step, the task was to create a model that predicts customer ratings with the highest possible accuracy. To do this, a first approach was to understand the objective and how to go about it. The problem addressed, which is the prediction of the star score of reviews, is a classification task that consists of classifying reviews into five categories ranging from a score of one to five based on the specific properties or attributes of each review. The classification of textual data is one of the most complex tasks in automatic natural language processing due to the many factors involved, namely the quality of the data cleaning, the algorithm used to represent the textual data, the quality of the data used to train the model, the prediction model used and its parameters, etc. As we can see, there are many factors and variables and the objective is to optimize each of these factors to obtain the best possible result.

What customers think is important information, it is very valuable knowledge for the organization concerned in the sense that it gives a very clear idea of the quality of a certain product or service and how the public perceives it. While some platforms allow users to give a star rating, most social networks do not offer this possibility of rating on a scale of one to five stars. The objective of this task will be to create a machine learning model that will be able to rank our data observations by assigning a star rating from 1 to 5 (Very Satisfied, Satisfied, Average, Dissatisfied, and Very Dissatisfied).

Once we have a numerical representation of our training dataset, we need to use classification algorithms, which are nothing more than supervised learning algorithms used to classify data observations into different categories. The goal at this point is to train classification models on our training dataset. The classification algorithm will identify patterns based on the characteristics of our training data observations and their corresponding labels, and the result of this identification will constitute our score prediction model. The models are expected to be generalized enough to predict classes for new data observations (without labels). Thus, having a generalized dataset is essential, as has already been pointed out.

Several learning algorithms were used for this task: SVM, KNN, Decision tree, Random forests and Logistic regression.

The model evaluation is an estimate that can be used to indicate how well you think the algorithm can actually perform. It is not a guarantee of performance since we are talking about a statistical estimate. Once we have estimated the performance of our algorithm, we can re-train the final algorithm on the training data set and prepare it for operational use. The performance of classification models is based on their ability to predict the outcome of new observations. This performance is measured against a test data set, which consists of observations that were not used to train the model. This textual dataset is a real post observation set that represents a real subset (which will be the test dataset of our model). This test dataset contains observations and their corresponding labels. The digital representation of the test dataset is fed into the already trained model and predictions are obtained for each observation. These predictions are then compared to the actual labels to determine the quality or accuracy of the model prediction.

In the result section we will detail the percentage of success of each model. Based on these results, we can observe that most of the models have a good performance, the multinomial logistic regression having the best performance and, on the other hand, the decision tree having the worst predictions.

V. Construction of a Sentiment Analysis Model

Sentiment analysis is one of the most active research areas in natural language processing. Sentiment analysis is a field of study that analyzes people's opinions, feelings, evaluations, attitudes and emotions through natural language.

The aim of sentiment analysis is to define a system that can extract subjective information (such as opinions and feelings) from natural language to create structured knowledge that can be used by decision support systems or decision makers. Nowadays, with the advent of social networks, sentiment analysis has gained in value. Their wide diffusion and their role in modern society represent one of the most interesting developments in recent years, attracting the interest of organizations and companies. Not so long ago, sentiment analysis was almost non-existent. Opinions were collected through surveys rather than through observation of textual data because computers were not capable of storing and processing large amounts of data, and there were no algorithms for extracting knowledge from written language.

The explosion of sentiment-laden content on the Internet, the increase in computational power, and advances in data mining techniques have made sentiment analysis a burgeoning research area and a crucial business sector. In this classification, we have implemented a recurrent neural network (RNN) with LSTM cells, where we will apply it on a digital representation of the data represented by One Hot Encoding [9], in order to classify the observations of the Post's customers into Very Satisfied, Satisfied, Average, Dissatisfied, and Very Dissatisfied opinions.

Recurrent neural networks are more effective than traditional neural networks and machine learning algorithms at retaining information from the previous event. The recurrent neural network involves a combination of loop networks. The loop network allows the information to persist. Each network in the loop takes the input and information from the previous network, performs the specified operation and produces an output, and at the same time passes the information to the next network. Some applications only need the most recent information, while other applications may need more information from the past, such as NLP (the meaning of related words). As the gap between the required prior information and the point of application has increased to a large extent, the learning of simple RNN lags behind. But fortunately, long-term memory networks [14], a special form of RNN, can learn such scenarios.

Long-term memory (LSTM) is an alternative architecture proposed in [14]: the traditional architecture of a Recurrent Neural Network (RNN) that is based on a simple activation function is modified in such a way that the vanishing gradient problem is explicitly avoided, while the learning method remains unchangeable. For more information on this architecture [6], [7]. But what are the strengths of LSTM ? why the LSTM will be effective to solve the steel continuous casting problem?

A LSTM neuron network is made up of several cell that have not just one activation function but rather three that are represented as an input gate, a forget gate and an output gate. Each cell remembers the state of the problem treat in several time intervals, and the three gates regulate the flow of information in and out of the cell. The LSTM network is very suitable for classification, processing and prediction based on time series data, because there may be lags of unknown duration between important events in the time series. This is what is needed to understand a series of words in a sentence. Also as we explained, LSTM was developed to deal with the explosion and disappearance of gradients that may be encountered when training traditional RNNs.

A. The Proposed LSTM Architecture and Hyperparameter

To define a neural network, it is necessary to establish parameters, such as training data, type of neural network, number of layers, connections, activation functions, propagation rules, etc.

There are several ways to train a neural network to produce a specific output for a specific input. In the current training method (Forward / Backward Propagation), we have error propagation, which involves adjusting the network based on each neuron's contribution to the error, and each neuron adapts the weights by using the gradient of descent. Genetic algorithms are also used to train neural networks [22]. By training these networks on a dataset with known correct outputs, the network will be able to return an approximate result for new data (not seen in the training stage). We trained our LSTM with data from 50,000 use cases (customer observations).

Regarding the architecture, there are several architectures proposed in the neural network language model with some differences between them. The proposed architecture for our application area consists of some basic principles, such as:

The input of our LSTM (the input sequence) is represented by a sentence, it is coded by the code from 1 to K, where K is the length of the client's observation. This involves the use of One Hot Encoding [9] to list individual words. The sequences are generated in 128 batches with 5% diversity (with the goal of avoiding overfitting).

The network topology is represented as follows: An input layer that takes as input the digital representation of the data observations. Five hidden layers have been implemented, each with 256 LSTM units, and each uses the "relu" activation function. For the five hidden layers, a normalization step is added via the Batch Normalization technique, which improves the performance and stability of neural networks. The main idea is to normalize the inputs to each layer so that they have an average output activation of zero and a standard deviation of 1. The regularization is used in the network in the form of an exclusion layer. This form of regularization prevents overfitting of the model. The output layer is composed of five neurons, as we want to perform a five-class classification with a "softmax" activation function that will be used to represent a probability distribution to predict customer sentiment. As shown, the neural network trains by adjusting the weights by comparing the results predicted by the neural network and the actual label of the observations in the dataset.

Now that we have built the model, we will have to train it on the digital representation of the training data. To do so, we will have to define a cost function which measures the difference between the results predicted by our model and the real results. As this is a classification case, we tested several functions but the experimental results show that the categorical function "cross-entropy" is the best for this application domain. So we have a neural network, a cost function to be minimized. To minimize the cost function, we used the gradient descent algorithm and exactly the Adam optimizer which is an optimization algorithm to minimize convex functions. As a learning standard, the used precision of the Adam optimizer is 0.001.

Finally, we define precision as an evaluation metric for our neural network. The following algorithm summarizes the main steps:

VI. Experimental Tests

In order to evaluate the performance of our models for the two treated part, this part puts forward the performance of the models taking into account the quality of the solution. Note that all the experiments were performed on the google colab under GPU.

A. Assessment of Classification Models

The evaluation metrics used for classifying textual data (observations of data with a star score ranging from 1 to 5) are generally as follows:

Accuarcy: is the sum of true negatives and true positives divided by the total number of observations.

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

Algorithm 1 : The Proposed LSTM for Better Customer Relationship Management.

```
- Data preparation pipeline (see section 3).
- Creation of the LSTM architecture : Creating eight layers, one input layer, five hidden layers each with 256 LSTM units, dropout layer for regularization and final output layer consists of five neurons and "softmax" activation function is used to predict the customer feelings (Very Satisfied, Satisfied, Fair, Dissatisfied and Very Dissatisfied).
- According to the unified law, the weights are initialized randomly.
- Codage : list all customer's observations. The One Hot Encoding is used to represent each observation and coded it.
- The categorical "cross-entropy" is used as cost function.
while index $\leq$ Max_iter do
    1. Five percent of diversity is generated in each batch.
    2. Measuring the difference between the results predicted by our model and the actual results, using cost function.
    3. Adam optimizers is used with a precision of 0,001 in order to minimize the cost function.
    4. Update the weights.
end while
```

Where TN = True Negatives, TP = True Positives, FN = False Negatives and FP = False Positives.

Precision: is the number of real positives on the set of positive cases predicted by the model.

$$Precision = \frac{TP}{TP + FP}$$

Recall: allows the performance of the model to be evaluated from the point of view of the positive class. It indicates the percentage of actual positive cases that the model is able to predict correctly out of the total number of positive cases.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score: a combination of "Precision" and "recall".

we can observe that most of the models have a good performance (a cause of the proposed data processing pipeline), the multinomial logistic regression having the best performance and the decision tree having the worst performance (Table I).

TABLE I. Metrics Associated with the Application of Models on the Data Represented by TF-IDF.

Algorithmes	Accuracy	Precision	Recall	F1 Score
Random Forst	0.8487	0.7203	0.8487	0.7792
Decision Tree	0.6513	0.7971	0.6513	0.7792
Multinomial Logistic Regression	0.8553	0.8017	0.8553	0.8257
SVM	0.8289	0.7952	0.8289	0.8079

Learning models are parameterized so that their behavior can be tailored to a given problem. Models can have many parameters, and finding the best combination of parameters can be treated as a search problem. To treat a search problem, we can use different search strategies to find a parameter or a robust set of parameters for an algorithm on a given problem, namely grid search.

Grid search is a parameter tuning approach that allows you to systematically build and evaluate a model for each combination of specified algorithm parameters. To perform Grid Search tuning to find the best parameters, we use the *GridSearchCV* class, but it is important to remember that this task consumes a lot of computing resources and takes a long time to return results.

We specify a few parameters for our model to experiment with, we can choose any values that make sense and the grid search model will return the best set of parameters.

Once our parameters are ready, all we have to do is initialize a gridsearch object and use it. To do this, we execute the parameters we have prepared and also specify the model and the metric that will be used to choose the best model. Once the gridsearch model is ready, we apply it to our data. We repeat this same step for the rest of the models and visualize the results as shown in Fig. 4. We can see that the multinomial logistic regression has the best performance.

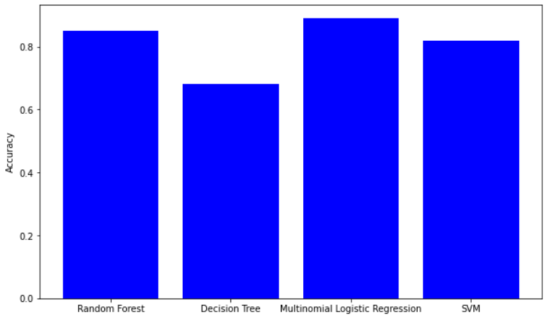

Fig. 4. Diagram of the Accuracy Metric of the Models after Applying the Gridsearch.

B. LSTM Network Assessment

To evaluate the RNN model with LSTM cells, we divided our data set (50,000 observations) into two sets, one for training and one for validation, and used cross-validation for training.

1) The Learning and Validation of LSTM Model: The learning rate gives an idea on the improvement of the quality of learning on a model. In Fig. 5, we present a graph that represents the learning rate (96%) and the validation rate (89%) of the proposed LSTM.

As can be seen from the model validation accuracy visualization, our RNN model with LSTM cells performed well and the fact that there is not a large discrepancy between the learning accuracy and the validation accuracy allows us to conclude that we do not have an overfitting (Fig. 6). We have also to mention that the learning rate (96%) and the validation rate (89%).

By comparing the accuracy parameters of LSTM with those of machine learning models, we can conclude that the learning rate is higher for LSTM, which will lead to good results. To test our model, we tested it with new and real data (not used in training). For this, we retrieved 1000 new observations (customer reviews with different classes) and compared the predictions with customer satisfaction (target provided by customers). The following confusion matrix (Fig. 7) shows the results of the test where the accuracy mitrics is equal to 94%.

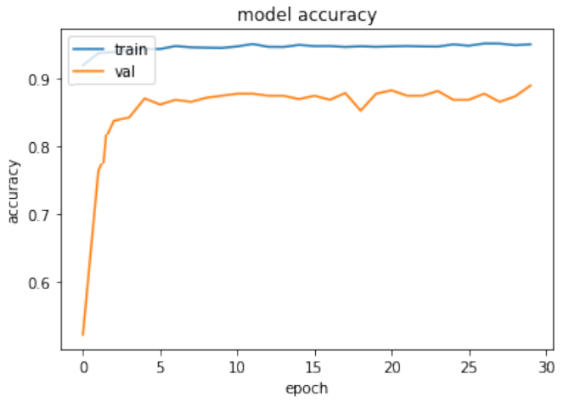

Fig. 5. Development of the Training and Validation Score per Epoch.

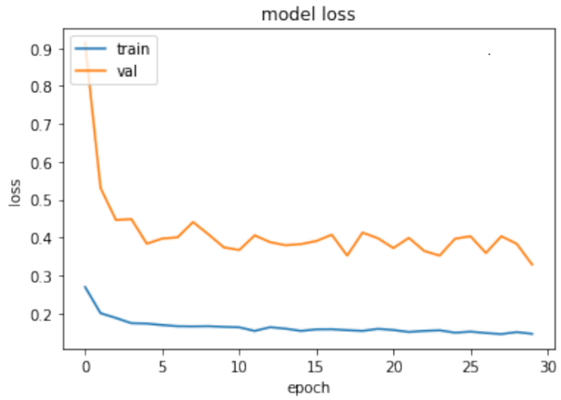

Fig. 6. Convergence of the Cost Function per Epoch

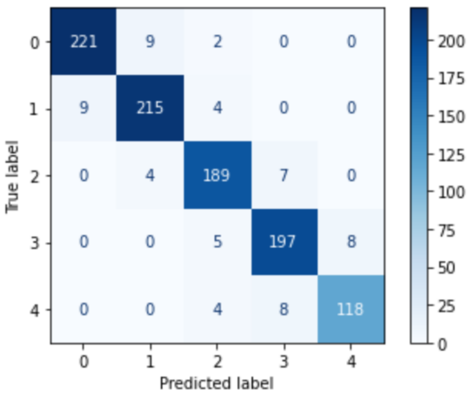

Fig. 7. The Confusion Matrix of the Test Set.

VII. Conclusion

In this paper, we have implemented many tasks to propose a data preparation pipeline ranging from data processing and cleaning to digital representation of textual data. We also performed Feature Engineering by digitally representing the text data using different algorithms.

The training data was collected from several companies operating in the same field as the position and the data set had to be balanced as much as possible, which was a challenge. Once the training database was formed, different models were prepared (SVM, KNN, decision tree, random forests, and logistic regression), in order to find the most effective model to rank the observations in our data by assigning them a star score ranging from one to five. After developing the score prediction model, we turned to developing a sentiment analysis model using RNN with LSTM. The performance of the proposed LSTM is very interesting such that the success percentage is 96%.

One of the future works we plan to develop is to generalize the approach on a GPUs cluster platform in order to process more complex documents and not only sentences.

Acknowledgment

The author would like to thank the Editor-in-chief and anonymous reviewers for their comments and suggestions to improve the quality of the paper.

References

[1] Armentano M. G., Godoy D., Campo M. and Amandi, A. NLP-based faceted search: Experience in the development of a science and technology search engine. *Expert systems with applications*, *41(6)*, 2886-2896, (2014).

[2] Bafna P., Pramod D. and Vaidya A. Document clustering: TF-IDF approach.*In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, *IEEE*, 61-66, (2016).

[3] Bagui S., Wilber C. and Ren K. Analysis of Political Sentiment From Twitter Data. *Natural Language Processing Research*, *1(1-2)*, 23-33,(2020).

[4] Bakarov A. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*, (2018).

[5] Balakrishnan V. and Lloyd-Yemoh E. Stemming and lemmatization: a comparison of retrieval performances, (2014).

[6] Berrajaa A. and Ettifouri E. H. The Recurrent Neural Network for Program Synthesis. *In International Conference on Digital Technologies and Applications*, *Springer, Cham*, 77-86, (2021).

[7] Berrajaa A. Solving the Steel Continuous Casting Problem using an Artificial Intelligence Model. *International Journal of Advanced Computer Science and Applications*, *vol. 12*, no 12, (2021).

[8] Bonaccorso G. Machine learning algorithms. *Packt Publishing Ltd*, (2017).

[9] Buckman J., Roy A., Raffel C. and Goodfellow I. Thermometer encoding: One hot way to resist adversarial examples. *In International Conference on Learning Representations*, (2018).

[10] Chowdhury G. G. Natural language processing. *Annual review of information science and technology*, *37(1)*, 51-89, (2003).

[11] Church K. W. Word2Vec. *Natural Language Engineering*, *23(1)*, 155-162, (2017).

[12] Eckhardt R. and Bagui S. Convolutional Neural Networks and Long Short Term Memory for Phishing Email Classification. *International Journal of Computer Science and Information Security*, *19(5)*, (2021).

[13] Hirschberg J. and Manning C. D. Advances in natural language processing. *Science*, *349(6245)*, 261-266, (2015).

[14] Hochreiter S. and Schmidhuber J. Long short-term memory. *Neural computation*, *9(8)*, 1735-1780, (1997).

[15] Hunt J. Regular expressions in python. *In Advanced Guide to Python 3 Programming*, *Springer, Cham*, 257-271, (2019).

[16] Jain A. K., Sahoo S. R. and Kaubiyal J. Online social networks security and privacy: comprehensive review and analysis. *Complex and Intelligent Systems*, 1-21, (2021).

[17] Janiesch C., Zschech P. and Heinrich K. Machine learning and deep learning. *Electronic Markets*, 1-11, (2021).

[18] Junnarkar A., Adhikari S., Fagania J., Chimurkar P. and Karia D. E-Mail Spam Classification via Machine Learning and Natural Language Processing. *In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, *IEEE*, 693-699, (2021).

[19] Kang Y., Cai Z., Tan C. W., Huang Q. and Liu H. Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, *7(2)*, 139-172, (2020).

[20] Kauffmann E., Peral J., Gil D., Ferrandez A., Sellers R. and Mora H. A framework for big data analytics in commercial social networks: A case study on sentiment analysis and fake review detection for marketing decision-making. *Industrial Marketing Management*, *90*, 523-537, (2020).

[21] Koga M., Mine R., Sako H. and Fujisawa H. Lexical search approach for character-string recognition. *In International Workshop on Document Analysis Systems*, *Springer, Berlin, Heidelberg*, 115-129, (1998).

[22] Lamos-Sweeney J. D. Deep learning using genetic algorithms. *Rochester Institute of Technology*, (2012).

[23] Lende S. P. and Raghuwanshi M. M. Question answering system on education acts using NLP techniques. *In 2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave)*, *IEEE*, 1-6, (2016).

[24] McCarthy, J. What is artificial intelligence?, (2007)

[25] Nugroho K. S., Sukmadewa A. Y. and Yudistira N. Large-scale news classification using bert language model: Spark nlp approach. *In 6th International Conference on Sustainable Information Engineering and Technology*, 240-246, (2021).

[26] Ritter A., Clark S. and Etzioni O. Named entity recognition in tweets: an experimental study. *In Proceedings of the 2011 conference on empirical methods in natural language processing*, 1524-1534, (2011).

[27] Watkins H., Gray R., Jha A. and Nachev P. An artificial intelligence natural language processing pipeline for information extraction in neuroradiology. *arXiv preprint arXiv:2107.10021*, (2021).

[28] Zhang Y., Jin R. and Zhou Z. H. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, *1(1-4)*, 43-52, (2010).