**Comparative study of traditional versus Deep learning LSTM algorithms for**

**Sentiment Analysis of Arabic text.**

Dissertation submitted

for the degree of

MSc Data Analytics 2019-2020

At Dublin Business School

**Krishan Goyal**

Supervisor(s): Terry Hoare

# Declaration

I, Krishan goyal, student of Dublin business School hereby declare that this research is my original work and that it has not been presented to any other institution/university for the award of Degree or Diploma. Also, I have also correctly referenced all literature and sources used in this work and this work is fully compliant with the Dublin Business School’s academic honesty policy.

**Signed**: Krishan goyal

**Date:** 5th January 2020

# Acknowledgements

I wish to express my sincere gratitude to Ms. Terri Hoare, my professor and research supervisor for her patient guidance from the start, valuable and constructive suggestions provided throughout the planning and development of my research, and her willingness to give her time generously, has been very much appreciated. I would also like to thank Mr. John O’Sullivan, my professor for his guidance and assistance in helping me with the pre-processing of the data and offering ample resources in the field of sentimental analysis. Finally, I wish to thank my family for their support and encouragement throughout my course.

# Abstract

This research compares the performance of traditional statistical and machine

learning algorithms with deep-learning models like LSTM and Bi-LSTM. Sentiment analyses are a crucial part of a firm’s operations. It aims at predicting and estimating the customer’s sentiments by correctly analysing the reviews provided by users and predicting their polarities to aid the decision making process. The current research uses text in Arabic dialect containing reviews about hotels that has been scraped from TripAdvisor website. Data pre-processing required for natural language processing is completed and base comparative models are built in Rapidminer for comparison against deep-learning using LSTM AND Bi-LSTM models using keras, tensorflow framework and python programming language.

# Table of Contents

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No.** | **Title** | | **Page No.** |
|  | **Declaration** | | **2** |
|  | **Acknowledgements** | | **3** |
|  | **Abstract** | | **4** |
|  | **List of Figures** | | **6** |
|  |  | |  |
| **1.** | **Introduction** | | **7** |
|  | 1.1 | Business Problem | 7 |
|  | 1.2 Research Problem | | 8 |
|  | 1.3 | Scope | 9 |
|  | 1.4 Limitations | | 9 |
|  | 1.5 | Dissertation Roadmap | 9 |
| **2.** | **Literature Review** | | **10** |
| **3.** | **Research Methodology and Methods** | | **14** |
| **4.** | **Data Analysis /Discussion** | | **31** |
|  | 4.1 | Exploratory Data Analysis | 31 |
|  | 4.2 Model Results | | 32 |
| **5.** | **Conclusion** | | **35** |
|  | 5.1 Conclusion | | 35 |
|  | 5.2 | Future works | 35 |
| **6.** | **Plagiarism and Referencing** | | **37** |
|  | **Appendices** | |  |
|  | **List of Abbreviations** | |  |

# List of Figures

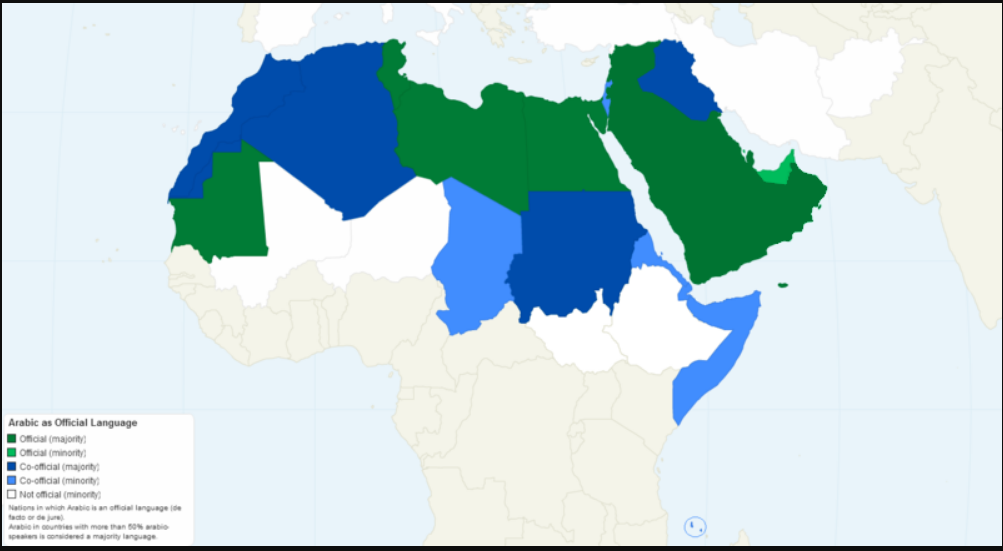
|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1.1 | Arabic speaking countries | 7 |
| 1.2 | Dissertation roadmap | 9 |
| 3.1 | CRISP-DM Methodology | 14 |
| 3.2 | Data preview | 16 |
| 3.3 | Sentiment analyses architecture | 16 |
| 3.4 | RapidMiner preview | 21 |
| 3.5 | Recurrent neural network | 22 |
| 3.6 | Recurrent neural network | 22 |
| 3.7 | Recurrent neural network | 23 |
| 3.8 | Recurrent neural network | 23 |
| 3.9 | LSTM architecture | 24 |
| 3.10 | LSTM architecture | 25 |
| 3.11 | LSTM architecture stepwise | 25 |
| 3.12 | LSTM architecture stepwise | 26 |
| 3.13 | LSTM architecture stepwise | 26 |
| 3.14 | LSTM architecture stepwise | 27 |
| 3.15 | LSTM architecture stepwise | 28 |
| 3.16 | Bi-LSTM Architecture | 28 |
| 4.1 | WORDCOUD of polarities | 31 |
| 4.2 | Pie chart of balanced data | 31 |
| 4.3 | Accuracy results of machine learning models | 32 |
| 4.4 | ROC Curve | 33 |
| 4.5 | AUC Results | 33 |
| 4.6 | BI-LSTM results | 34 |
|  |  |  |

**Chapter 1 — Introduction**

* 1. **Business Problem**

In past few years, Sentiment analysis has been the focus of many research studies due to the wide variety of its potential applications. Many of these studies have relied heavily on available resources mostly in the form of polarity annotated data sets or sentiment lexicons such as SentiWordNet.

At the same time, the Arabic language has shown rapid growth regarding its users on the internet, moving up to the 4th place in the world ranking of languages by the users according to internetworldstats. Arabic and its various vernaculars are spoken by around **422 million speakers** (local and non-local) in the Arab world just as in the Arab diaspora making it one of the five most communicated in dialects on the planet. As of now, **40 nations** are part conditions of the Arab League (just as 5 nations were allowed an eyewitness status) which was established in Cairo in 1945.



1.1

Arabic is a language group containing 30 or so current assortments. This, along with the major happenings in the Middle East, show a large potential for sentiment analysis and consequently an urgent need for more reliable processes and resources for addressing it. Because of that, there has been an increasing interest and research in the area of Arabic sentiment analysis.

However, The Arabic Language remains under resourced with respect to available data. This can be attributed to the fact that most resources developed within studies addressing Arabic sentiment analysis, are either limited in size, not publicly available or developed for a very specific domain. With the development of web based life, a lot of significant information become accessible on the web and simple to get to. Web based life clients examine all that they care about through blog entries or tweets, share their suppositions and show intrigue unreservedly; while they don't really do it face to face. We read about political discussions, social issues, inquiries regarding a specific item, and so on. Organizations likewise utilize interpersonal organizations to advance their items and benefits, and investigate individuals' sentiments to improve their items and administrations, subsequently creating an enormous measure of information. In this specific situation, the requirement for a systematic device that can procedure the client's information and arrange them in regards to opinion polarities is expanded and become a need.

Sentimental analyses (SA) or Opinion Mining (OM) is the undertaking of deciding and recognizing the extremity/supposition in a given bit of content and characterizing positive or negative. English and other European dialects have been investigated in most of SA devices yet hardly any ongoing examination endeavours stretch out the concentration to other low-assets dialects, for example, Arabic and provincial Arabic.

The situation in Arabic language is an exemplary instance of Diglossia, whereby the composed proper language contrasts significantly from the verbally expressed vernacular. modern standard Arabic (MSA) is vigorously founded on Classical Arabic and establishes the authority composed language utilized in government undertakings, news, communicate media, books and instructions. modern standard Arabic goes about as the most widely used language among Arabic local speakers. Notwithstanding, the communicated in language (by and large alluded to as Dialectal Arabic) broadly changes over the Arab world. In addition, there is neither standard composed orthography nor formal sentence structure for these vernaculars. To foresee the estimation of an Arabic bit of content, most of the works depend on Machine Learning (ML) calculations like Linear Support Vector Classification (Linear SVC), Multinomial Naive Bayes (MNB) and others. Despite the fact that these classifiers are direct to execute and accomplish great outcomes, they require a ton of highlights building before applying the information to the classifiers. Along these lines, work in Arabic supposition examination still depends vigorously on the morphological and syntactic parts of the language, for example, POS labelling, word stemming, the notion vocabularies and other hand-made highlights, yet after the noteworthy improvement realized by Deep Learning (DL) over the customary ML draws near, scientists will in general research and investigate the exhibition of profound neural systems in breaking down various types of Arabic messages and concentrate highlights for some NLP undertakings, for example, Language Identification, Text Summarizing, Sentiment Analysis, etc.

In this paper, we introduce a deep neural network which uses **Bi-Directional** **Long-Short Term Memory Networks (Bi-LSTM)** which is an improvement over recurrent Neural Net- works (**RNN**) and simple LSTMs to predict the polarity of a text and classify it as either having positive or negative polarity.

**1.2 Research Problem**

Sentimental analysis is a crucial part of a firm’s operations.it is a decision-making tool. Sentimental analysis aims at predicting and estimating the future demand of products by analysing the data and determining and detecting the polarity/opinion of users and classifying it

into positive or negative. This research is conducted using different Natural language processing (NLP) and Deep learning algorithms to predict the polarity of sentiments among Arabic speakers about different products.

**Research Question**: **Sentimental analysis of Arabic dialect data using deep learning, Natural language processing and long short term memory models.**

**Aim**: Accurately analysing the sentiments among Arabic speakers to predict the future demands for each of the product to aid managerial decisions.

**Objective**: To compare the best statistical models and machine learning and deep learning algorithms to predict the correct sentiments in the text provided in Arabic dialect, containing reviews about hotels, that is scraped from TripAdvisor website and the polarities (positive or negative) of these reviews to better understand the Arabic user’s sentiments.

**1.3 Scope**

The scope of this research is to help build a sentimental analysing tool that includes a wide range of Machine learning and deep learning algorithms to accurately predict the sentiments among Arabic speakers and writers and help in better understanding of Arabic people and language

**1.4 Motivation for our research**

Much research is conducted on subjectivity and sentiment analysis of English language testing during the last decade but compared to English, other languages are more difficult to process and hence are less studied upon.

Very less work is focused on Arabic text and getting required datasets for data analyses process for NLP is difficult and challenging at the same time. In our research we will be using a dataset in which data is scraped from TripAdvisor website about hotel reviews.

The work on Arabic is mostly focused on modern standard Arabic or MSA but Most of the comments are in colloquial Arabic i.e. common language which is a mix of dialects and is difficult to process than MSA.

**1.5 Dissertation Organisation**

To create a strategic plan for implementing the dissertation project, the following roadmap is used.

Introduction

This chapter includes the problem definition, research question, aim and objectives and the hypothesis to be

tested.

Literature

Review

This chapter highlights the existing research for time series forecasting with the use of research journals and

books which include the theories, concepts and models of forecasting.

Methodology

This chapter utilizes the CRISP

-

DM approach to conduct the research, each of the six phases of the

methodology are tailored for this research.

Data

Analysis

The aim of this chapter is to compare the finding and the performance of each algorithm without concluding

the finding.

Discussions

This chapter includes the interpretation of the algorithms' results, discussion of the findings and answering

the research question.

Conclusion

This chapter will summarise the finding of the research to come to a conclusion.

## Chapter 2 – Literature Review

The current research done for time series forecasting is investigated to identify the best performing forecasting methods for both Statistical and Machine learning algorithms. It is a difficult task to conclude which are the best models (Traditional statistical models, machine learning algorithms or deep learning algorithms) for forecasting time series data. To better understand the theories, concepts and models required for Demand forecasting, the following Books and Research journals were referred.

**Chapter 2 Literature Review**

In the field of sentiment analysis, different machine learning approaches have been applied. Feature determination task engaged with machine learning classification as a technique to improve the presentation by decreasing the dimensionality. In the English language, there exist a considerable amount of published work that have used different methods of feature selection to improve the performance of sentiment analysis but, little consideration has been given to feature selection impact on provincial Arabic slant examination. Sentiment Analysis is normally viewed as a supervised classification task, where the data are ordered into at least two opinions classes by giving a dataset with the text and the sentiment label. The common approach in Sentimental Analysis is the usage of Machine Learning through language modelling and feature engineering. The majority of the Sentimental Analysis methods in Arabic use words and character n-gram feature with various portrayal settings and various classifiers. In some cases, there are extra and significant wellsprings of features that have been utilized to enhance features for Sentimental Analysis.

Building sentiment analysis resources for the Arabic language, has been addressed by a number of researchers. For sentiment annotated corpora Rushdi-Saleh et al. presented OCA; a dataset of 500 annotated movie reviews collected from different web pages and blogs in Arabic. Although the dataset is publicly available, it is limited in size and only covers the movie reviews domain.

Abdul-Mageed & Diab presented the AWATIF multi-genre corpus of Modern Standard Arabic labelled for subjectivity and sentiment analysis. The corpus was built from different resources including the Penn Arabic Treebank, Wikipedia Talk Pages and Web forums. It was manually annotated by trained annotators and through crowd sourcing. The dataset targets only Modern Standard Arabic which is not commonly when writing reviews on most websites and social media. Moreover, the dataset is not available for public use.

LABR a large dataset of 63K, polarity annotated, Arabic Book reviews scrapped from www.goodreads.com. On this site, each review is rated on a scale of 1 to 5 stars which the authors have mapped to a sentiment polarity. The dataset was then used for the tasks of sentiment polarity classification and rating classification. The large scale dataset is publicly available for use; however, it only covers the domain of book reviews. For sentiment lexica, as a part of a case study exploring the challenges in conducting sentiment analysis on Arabic social media, El-Beltagy et al. [5] developed a sentiment lexicon including more than 4K terms. The lexicon was semi-automatically constructed through expanding a seed list of positive and negative terms by mining conjugated patterns and then filtering them manually.

El-Sahar and El-Beltagy exhibited a completely computerized way to deal with separate dialect sentiment slant vocabularies from twitter streams utilizing lexicon-syntactic examples and point wise mutual information.

All the more as of late, SANA, a huge scale multi-sort sentiment lexicon was introduced. SANA is made up of 224.5k entries covering Modern Standard Arabic MSA, Egyptian Dialectal Arabic and Levantine Dialectal Arabic.

SANA is built from different resources including The Penn Arabic Treebank, Egyptian chat logs, YouTube comments, twitter and English SentiWordNet. Some of the lexicon components were built manually, others were obtained using automatic methods such as machine translation. Various techniques were used to evaluate the generated lexicon. The lexicon is not publicly available. In [12], the authors introduce a subjectivity and sentiment analysis system for Arabic tweets by extracting different sets of features such as the form of the

words (Stem, Lemma), POS tagging, the presence of the sentiment adjective and the Arabic form of the tweet (Modern Standard Arabic or Data Analysis), in addition to other Twitter-specific features such as the user ID (person, organization) and the gender of the user.

In [10] a language model is built and different machine learning classifiers are used

to handle tweets in Modern Standard Arabic and Jordanian. An early deep learning framework for Sentiment Analysis for Arabic is proposed in [13]. The authors explore several network architectures based on Deep

Belief networks, Deep Auto Encoder and the Recursive Auto Encoder. The authors there do not mention the range of labels of the polarity classification. They use The Linguistic Data Consortium Arabic Tree Bank dataset and show that the model outperforms the state of the art models on the same dataset by around 9% in terms of F-score. They get an accuracy of 74.5 percent.

Baly et al. fabricate a profound learning model to distinguish the polarities of tweets in a 5-scale order that extents from extremely negative to positive. They recover tweets from 12 Arab nations in 4 locales (the Arab Gulf, the Levant, Egypt and North Africa). They gather 470K tweets. Their profound learning model comprises of an inserting layer pursued by a Long Short Term Memory layer. Pre-prepared word implanting's are applied utilizing the skip-gram model from Word2Vec. The creators examine the presentation of their model on various morphological structures (lemma and stem). They accomplish an exactness of 70% for the Egyptian tweets and lemma installing's while for UAE tweets they get 63.7% precision.

Soumeur et al. research the Sentiment Analysis in the Algerian clients' remarks on different Facebook brand pages of organizations in Algeria. They gather 100000 remarks written in Algerian, however they just comment on 25000 remarks as positive, negative or unbiased. They apply a Convolutional Neural system as an element extractor and change organize. Their model comprises of three sort of layers, three Convolutional Neural system layers each with 50 channels and 3 portion size, trailed by pooling layers and the completely associated layers to anticipate the feeling of the remark. Their model accomplishes a 89.5 percent precision. SEDAT, a wistful and enthusiastic analyser model, was worked in [16] utilizing Arabic tweets. Word embeddings and archive embeddings notwithstanding a lot of semantic highlights are utilized. All the removed highlights into Convolutional Neural system – Long Short Term Memory systems pursued by a completely associated layer are applied. The information was gotten from the open datasets for SemEval 2018, which has a size of almost 7000 tweets. The creators further compute Spearman's relationship coefficient over the pattern models which they outflank with 0.01- 0.02 purposes of contrast. Recently, an ensemble deep learning model was proposed. There, the authors combine Convolutional Neural network and – Long Short Term Memory models to predict the sentiment of Arabic tweets exploiting Arabic Sentiment Tweets Dataset. The model outperforms the state-of-the-art deep learning models F1-score of 53.6 percent, as they achieve an accuracy of 65percent and an F1-score of 64.46 percent.

Abbasi et al. built up a hybridized genetics calculation that consolidates the IG heuristic for feature selection called Entropy Weighted Genetic Algorithm. Their technique was intended to assess diverse capabilities comprising of syntactic and elaborate features for English and Arabic dataset. They applied their proposed strategies and highlights to a multi-language web discussion at the archive level. They announced that utilizing EWGA with Support Vector Machine acquired superior levels, with accuracy of over 93 percent for the Arabic language.

Duwairi and El-Orfaili explored the impact of feature correlation on Arabic slant sentiment classification performance execution. Distinctive N-gram models of words and characters were utilized for text representation portrayal. Both Modern Standard Arabic and common Arabic lingo were available in the dataset that incorporates political and movie reviews and surveys. Three classifiers were utilized to characterize the audits, to be specific, SVM, Naïve Bayes, and K-NN. Critical improvement has been acquired with applying the main 1.2k related features and word N-gram utilizing Naïve Bayes classifier. SVM and Naïve Bayes classifiers showed better performance, where Naïve Bayes classifier yielded the highest accuracy with 97.2 percent.

Shoukry and Rafea proposed a hybrid methodology for sentiment classification of Egyptian Arabic dialect. The feature vector was worked of various N-gram models and extremity scores. For diminishing the features, they doled out a recurrence edge for each component through 20k tweets. They used an SVM classifier to train the data.

The outcomes indicated that the best execution is 84 percent, and got with a blend of uni gram, bi gram, and tri gram model.

[20] introduced in their work an exploratory assessment to take a gander at the effect of different content portrayal conspires on machine learning strategies for Arabic semantic estimation investigation. They explored different avenues regarding different datasets which contain tweets written in MSA, Egyptian vernacular, and Saudi tongue. Text representation schemes on machine learning 3 classifiers were chosen, namely, Support Vector Machine, compliment NB, and multinomial NB. For selecting the optimal features, they employed IG method and other basic data pre-processing techniques. Be that as it may, IG didn't achieve a critical improvement in execution. The outcome indicated that a blend of uni-gram and bi-gram acquired the most elevated precision with the IDF weighting plan.

[21] explored the significance of utilizing a hereditary calculation include determination approach in Arabic slant examination. They utilized LABR dataset which contains Modern Standard Arabic and colloquial surveys about books. The experiment was performed with five Machine Learning supervised algorithms Support Vector Machine, Naive Bayes, Multinomial Naive Bayes, Stochastic Gradient descent and Decision Trees. As noted from the outcomes, the accuracy improved subsequent to applying the genetic calculation, where MNB yielded the most imperative exactness (94 percent) with uni-grams and Term Frequency-Inverse Document Frequency weighting

[22] introduced a cross breed approach for sentiment analysis of Modern Standard Arabic and Egyptian Dialectal tweets. They utilized Information Gain strategy to choose the pertinent features which have been bolstered to Support Vector Machine and Random Forest classifiers. Just as, different assessment highlights were utilized dependent on various dictionaries. In light of their outcomes, the methodology got 90 percent accuracy.

Omar et al. [25] presented an empirical comparison of seven feature selection methods (Information Gain, PCA, Relief-F, Gini Index, Uncertainty, Chi-square, and Support Vector Machines (SVMs) for Arabic sentiment analysis. They utilized three classifiers (Support Vector Machines, Naive Bayes, and K Nearest Neighbour) for Modern Standard Arabic and colloquial reviews about movie. Creators saw that a critical enhancement for the exhibition when highlight determination techniques are utilized. Their trial results indicated that Support Vector Machines classifier with Support Vector Machines -based element determination strategy acquired the most noteworthy accuracy with 92.4 percent.

In light of the reviews of numerous past examinations on Arabic estimation investigation, it is observed well that may be seen that the vast majority of the examinations have concentrated uniquely on the correlation between various classifiers Besides, a couple of studies have tended to the element determination techniques to determine the issue of the dimensionality in regional Arabic. Also, the feature choice strategies have been exclusively used and explored, without thinking about joining these techniques. The combination can exploit the advantages of the feature selection methods and avoid their disadvantages. Consequently, other than researching the impact of various individual strategies for feature selection, this work additionally analysed their blends on the regional Arabic assessment investigation. Also, the impact of various weighting plans, stemmers, and feature models on the exhibition was researched.

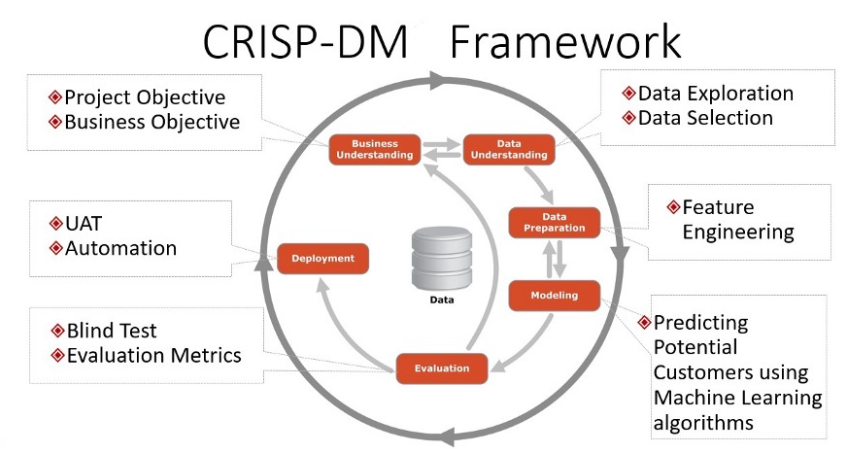
From the literature review it is evident that the current research conducted between different models are concluded based on the overall performance of those models. Many researchers have compared different statistical and machine learning models with deep learning models with LSTM’S and the results show that LSTM method had good accuracy among other statistical models and Natural Language Processing models like recurrent neural network, had good accuracy amongst other machine learning algorithms

**Chapter 3 - Research Methodology and Methods**

The research was conducted using the Cross-Industry Standard Process for Data Mining methodology (CRISP-DM). CRISP-DM is the most popular method for data analytics, data science and data mining projects which was developed in 1996 This methodology provides a structure for the research that aids better and faster results. CRISP-DM spreads out to six significant periods of the investigation procedure, with steps to be taken in every one of them. It’s not a linear process; the phases represent an ongoing cycle of action and analysis, and there’s often a lot of back and forth within and between phases.

, these phases help better understand the process and provides a road map to follow when planning and carrying out the research. The figure below shows the phases of the CRISP-DM model.

The arrows show the flow of the process and the frequent dependencies between the phases. The phases of the CRISP-DM methodology are as follows:



**3.1 CRISP-DM Methodology**

**STEPS: -**

1. Business Understanding

2. Data Understanding

3. Data Preparation

4. Modelling

5. Evaluation

6. Deployment

1. **Business Understanding**

This is the first phase of CRISP-DM which focuses on understanding the project objectives and requirements from a business perspective, i.e. having clear understanding of what the business requires to achieve from this project, background of the business and the success criteria of the project, then converting this knowledge into a data mining issue definition and a starter plan intended to accomplish the goals. The key steps involved in business understanding determining business objectives, assessing the situation, determining the data mining goals and lastly producing plans that are projected in achieving these goals.

Sentimental analysis is a crucial part of a firm’s operations.it is a decision-making tool. Sentimental analysis aims at predicting and estimating the future demand of products by analysing the data and determining and detecting the polarity/opinion of users and classifying it into positive or negative. This research is conducted using different Natural language processing (NLP) and Deep learning algorithms to predict the polarity of sentiments among Arabic speakers about different hotels. With the help of this process, business can analyse the sentiments of its users about their hotels services and their reaction to the customers according to the comments people post online and improve the quality of their products and services in future.

**2. Data Understanding**

The data understanding phase starts with an initial data collection from the all the sources which can help in achieving goals and proceeds with activities in order to get acquainted with the information, to distinguish data quality issues, to find first bits of knowledge into the information or to identify fascinating subsets to frame hypotheses for hidden information. It also includes data description report that describes the data in hand, data exploration and lastly verifying data quality.

Finding and extracting of Arabic reviewing content from the internet is considered to be a hard task relatively to English. This is due to the smaller number and activity of Arabic based e-commerce & reviewing websites over the internet, also that many Arabic speakers use the English language or Arabic transliterated in Roman characters to write their reviews. All this has had a big impact on reducing the amount of pure Arabic reviews on the internet. Fortunately, recently the Arabic reviewing content over the internet has shown a significant growth, moreover new reviewing websites has been built. In this study we make use of the available reviewing Arabic content over the internet to create a dataset reliable for the task of sentiment analysis.

We will be using a dataset containing data about reviews of hotels collected from multiple users of TripAdvisor community and sentiments of users according to their comments. This data was scraped from trip advisor page containing Arabic language reviews only. Data is divided into two polarities, 1 representing positive reviews and -1 representing negative reviews. Dataset contains 10775 numbers of positive reviews and 2765 number of negative reviews and two attributes 

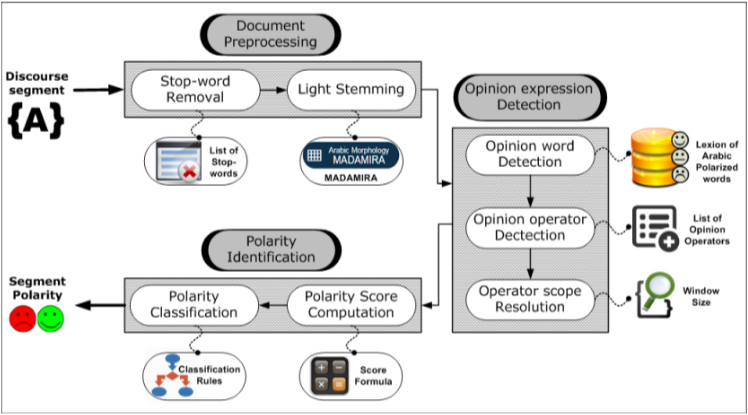
3.2

**TEXT**: Arabic comments from users, Reviews about hotels describing their experience in details.

**POLARITY**: The sentiments of comments and reviews classified as positive or negative.

**3. Data Preparation**

The data preparation stage covers all exercises to build the last data set from the underlying crude information. Few steps in this phases are data set description, selecting rationale data i.e. choosing the inclusion or exclusion of data, cleaning collected data, creating derived and generated records, merging data from all different sources and reformatting the data into required format for next steps.



3.3 Sentiment analyses architecture

A problem with modelling text is that it is messy, and techniques like machine learning algorithms prefer well defined fixed-length inputs and outputs.

Machine learning algorithms can't work directly with crude content text legitimately; the content must be changed over into numbers. This is called feature extraction or feature encoding. A well-known and straightforward strategy for feature extraction with content information is known as the Bag of-words model of content.

**Bag of-Words**

A Bag of words model, or BOW for short, is a method for extracting features from content for use in demonstrating, for example, with machine learning algorithms. The methodology is extremely simple, basic, flexible and adaptable, and can be utilized in a heap of ways for extracting features from documents. A bag of-words is a portrayal of content that depicts the event of words inside a record. It includes two things:

1. A known words vocabulary.

2. A measure of the presence of meaningful words.

It is known as a "Bag" of words, because of the fact that any data about the order or structure of words in the record is disposed of. The model is just focused about whether realized words occur in the archive, not where in the report its located.

In this methodology, we look at the histogram of the words inside the content, for example considering each word count as a feature. The instinct is that reports are comparative in the event that they have comparative substance. Further, that from the substance alone we can master something about the importance of the report. The bag of-words can be as basic or intricate as you prefer. The intricacy comes both in concluding how to structure the jargon of known words (or tokens) and how to score the nearness of known words.

**Managing Vocabulary**

As the size of vocabulary increments, so does the vector portrayal of the documents. For an extremely enormous corpus, for example, a large number of books, that the length of the vector may be thousands or millions which is a large number of positions. Further, each documents may contain not many of the known words in the vocabulary. This outputs in a vector with bunches of zero scores, called a sparse representation or sparse vector.

Sparse vectors require more memory and computational resources when modelling and the vast number of positions or dimensions can make the modelling process very challenging for traditional algorithms. As such, there is strain to diminish the size of the vocabulary when utilizing a bag of-words model.

There are basic content cleaning methods that can be utilized as an initial step, for example,

• Ignoring case

• Ignoring punctuation

• Ignoring frequent words that don't contain a lot of meaningful information, called stop words, similar to "an" ,"of”, “the" and so forth.

• Fixing misspelled words.

• Reducing words to their stem (e.g. “play” from “playing”) using stemming algorithms.

A more sophisticated approach is to create a vocabulary of grouped words. This changes both the extent of the vocabulary and enables the bag of words to catch somewhat additionally important meanings from the records.

In this methodology, each word or token is known as a "gram”. Creating a vocabulary of two-word pairs is, in turn, called a bi-gram model. Again, only the bi-grams that appear in the corpus are modelled, not all possible bi-grams.

An N-gram is an N-token sequence of words: a 2-gram (more commonly called a bi gram) is a two-word sequence of words like “please turn”, “turn your”, or “your homework”, and a 3-gram (more commonly called a tri-gram) is a three-word sequence of words like “please turn your”, or “turn your homework”.

For example, the bi-grams in the first line of text in the previous section: “It was the best of times” are as follows:

• “it was”

• “was the”

• “the best”

• “best of”

• “of times”

A vocabulary then tracks triplets of words is called a tri-gram model and the general approach is called the n-gram model, where n refers to the number of grouped words.

Often a simple bi-gram approach is better than a 1-gram bag of words model for tasks like documentation classification.

a bag of bi-grams representation is much more powerful than bag of words, and in many cases proves very hard to beat.

**Scoring Words**

Once a vocabulary has been chosen, the occurrence of words in example documents needs to be scored.one very simple approach to scoring: a binary scoring of the presence or absence of words. Some additional simple scoring methods include:

• Counts. Tally the occasions each word shows up in a document.

• Frequencies. Compute the recurrence that each word shows up in a report out of the considerable number of words in the record.

**Word Hashing**

You may recollect from software engineering that a hash function is a touch of math that maps information to a fixed size arrangement of numbers. For instance, we use them in hash tables when programming where maybe names are changed over to numbers for quick query. We can utilize a hash portrayal of known words in our vocabulary. This tends to the issue of having a huge vocabulary for a huge book corpus since we can pick the size of the hash space, which is thus the size of the vector portrayal of the report. Words are hashed deterministic-ally to a similar whole number record in the objective hash space. A parallel score or check would then be able to be utilized to score the word. This is known as the feature hashing.

it is also called hash trick. The test is to pick a hash space to suit the picked vocabulary size to limit the likelihood of impacts and exchange off sparsity.

**TF-IDF**

An issue with scoring word recurrence is that exceptionally visit words begin to command in the archive (for example bigger score), yet may not contain as much "educational substance" to the model as rarer yet maybe area explicit words.

One methodology is to rescale the recurrence of words by how regularly they show up in all archives, so the scores for visit words like "the" that are additionally visit over all reports are punished.

This way to deal with scoring is called Term Frequency – Inverse Document Frequency, or TF-IDF for short, where:

• **TF: Term Frequency**, which quantifies how regularly a term happens in a report. Since each report is distinctive long, it is conceivable that a term would show up substantially more occasions in long archives than shorter ones. In this manner, the term recurrence is regularly isolated by the report length (otherwise known as. the absolute number of terms in the archive) as a method for standardization:

TF(t) = (Number of times term t shows up in a report)/(Total number of terms in the record).

• **IDF: Inverse Document Frequency**, which quantifies how significant a term is. While registering TF, all terms are considered similarly significant. Notwithstanding, it is realized that specific terms, for example, "is", "of", and "that", may seem a great deal of times however have little significance. Hence we have to overload the continuous terms while scale up the uncommon ones, by processing the accompanying:

IDF(t) = loge (Total number of records/Number of reports with term t in it).

The scores are where not all words are similarly as significant or fascinating. The scores have the impact of featuring words that are particular (contain helpful data) in a given record. In this way the idf of an uncommon term is high, though the idf of a regular term is probably going to be low.

We will be using TF-IDF for our deep learning baseline model.

**Restrictions of Bag-of-Words**

The Bag of words model is exceptionally easy to comprehend and actualize and offers a great deal of adaptability for customization on your particular content information. It has been utilized with extraordinary accomplishment on expectation issues like language demonstrating and documentation grouping. In any case, it experiences a few deficiencies, for example,

• Vocabulary: The jargon requires cautious plan, most explicitly so as to deal with the size, which impacts the sparsity of the archive portrayals.

• Sparsity: Sparse portrayals are more enthusiastically to show both for computational reasons (existence intricacy) and furthermore for data reasons, where the test is for the models to bridle so little data in such an enormous illustrative space.

• Meaning: Discarding word request ignores the novel situation, and along these lines significance of words in the record (semantics). Setting and significance can offer a ton to the model, that at whatever point showed could separate between comparable words contradistinction organized ("this is interesting" versus "is this intriguing"), equivalent words ("old bicycle" versus "utilized bicycle"), and significantly more.

**SMOTE UPSMAPLING**

We propose an over-sampling approach in which the minority class is over-inspected by making "synthetic" models instead of by over-sampling with substitution. This methodology is propelled by a system that demonstrated effective in written by hand character acknowledgment (Ha and Bunke, 1997). They made additional preparation information by playing out specific activities on genuine information. For their situation, activities like skew and rotation were common approaches to annoy the preparation data.

We create engineered models in a less application-explicit way, by working in feature space instead of data space. The minority class is over-sampled by taking every minority class test and presenting engineered models along the line sections joining any or all entirety of the k minority class nearest neighbours.

Contingent on the measure of over-sampling required, neighbours from the k nearest neighbours are randomly picked. Our implementation currently uses five nearest neighbours. For instance, if the amount of over-sampling needed is 200%, only two neighbours from the five nearest neighbours are chosen and one sample is generated in the direction of each.

Synthetic samples are produced in the accompanying manner: Take the contrast between the feature vector (sample) viable and its closest neighbour. multiplying this distinction by an arbitrary number somewhere in the range of 0 and 1, and add it to the component vector viable.

This causes the determination of an irregular point along the line section between two explicit features. This methodology adequately powers the choice locale of the minority class to turn out to be progressively broad.

In our dataset, we had 13422 records of both polarities, out of these 13422 observations, only 2647 had negative polarity while the rest of 10775 observations were positive reviews about the hotels. We used smote up sampling model in rapid miner to balance our dataset. This process creates synthetic examples of minority class with help of k nearest neighbour model, with the help of this process we balanced our dataset to create equal number of observations for both of our polarities, thus increasing the negative observations in our dataset equalling to positive observations.

Moreover, we assessed the impact of stop words removal and stemming (light stemming and root stemming) on the presentation of the classifier. For the experimenting, list of stop words and the stemmers provided with Rapid miner were used.

**4. Modelling**

In this phase, various modelling techniques are selected according to the data collected and the result to be obtained, also the arguments needed for the models to be applied and their parameters are calibrated to optimal values. This model is then applied on test data sets to see if its working based on past results. Then the final model is build based on the test results and the model is assessed and its parameters are checked.

.

In this research, the deep learning and machine learning models are initially implemented as baseline models using rapid miner on the pre-processed training set to train the model from eighty percent of our observations. Then the model is tested on the remaining twenty percent of data. The results of these models are then compared with results we achieve by adding LSTM and Bi-LSTM layers. For implementing the Bi-LSTM model, we will be using the keras library in python for tensorflow framework, we will be also learning in detail why it is more efficient the Recurrent Neural networks and help overcome issues we face in common RNN networks

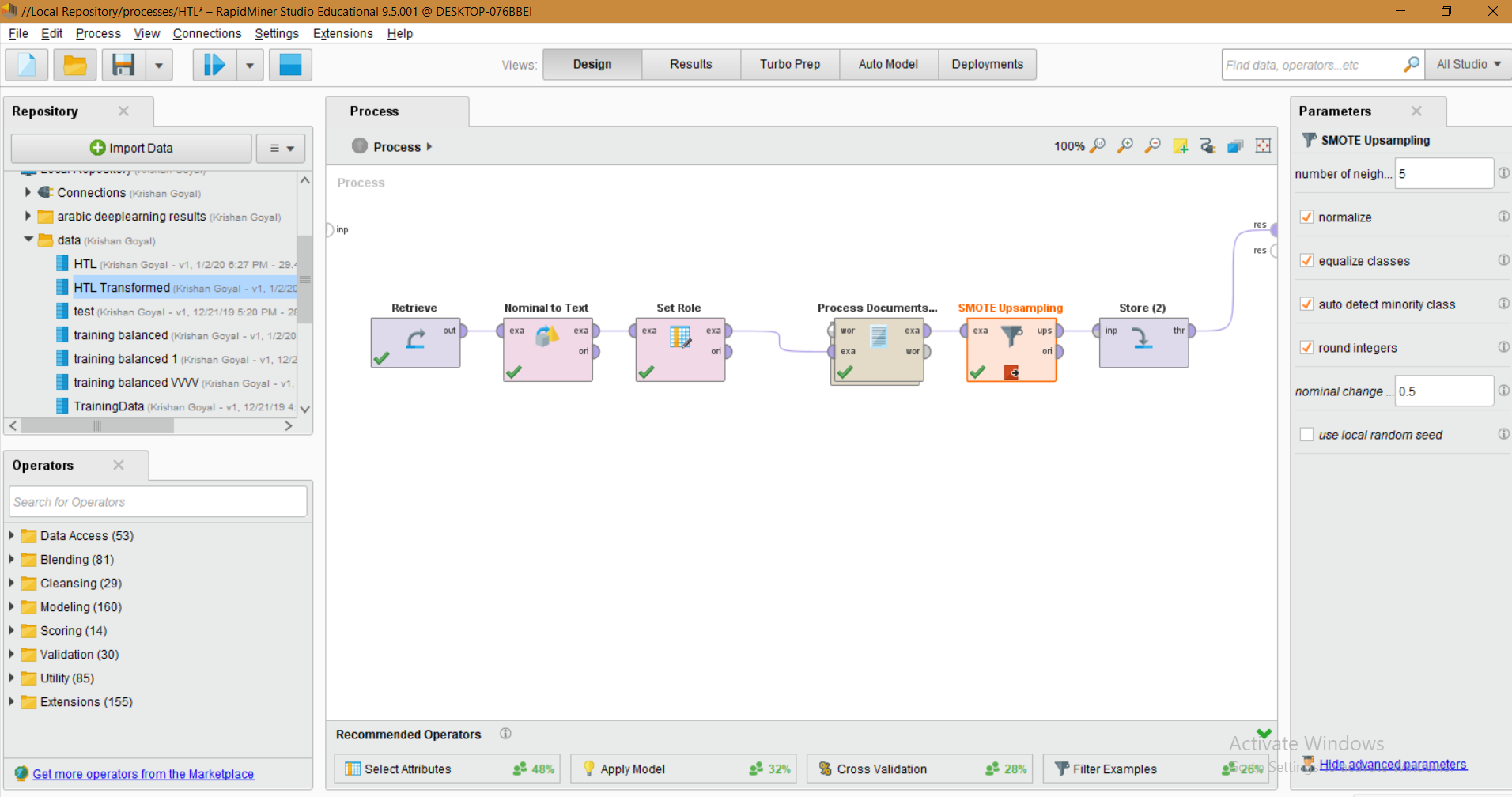
**RAPIDMINER**

RapidMiner is a software package that allows data mining, text mining and predictive analytics.

The program allows the user to enter raw data, including databases and text, which is then automatically and intelligently analysed on a large scale.

The basic process is drag and drop and lets you build out models and solutions very quickly. You can coordinate python and R on the off chance that you have extra preparing that you need to perform.

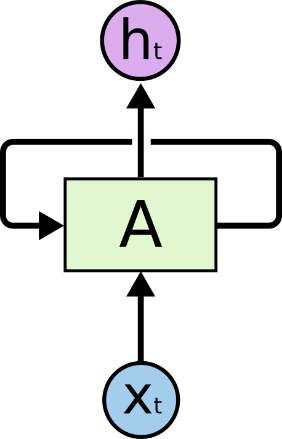
We will be using rapidminer in this research to create a baseline deeplearning model for our sentimental analysis problem. The results and the process created will then be compared to our LSTM models results.



3.4 Rapid miner UI

**Recurrent Neural Networks**

Recurrent neural systems are systems with loop in them, enabling data to persevere and transfer to the next step.

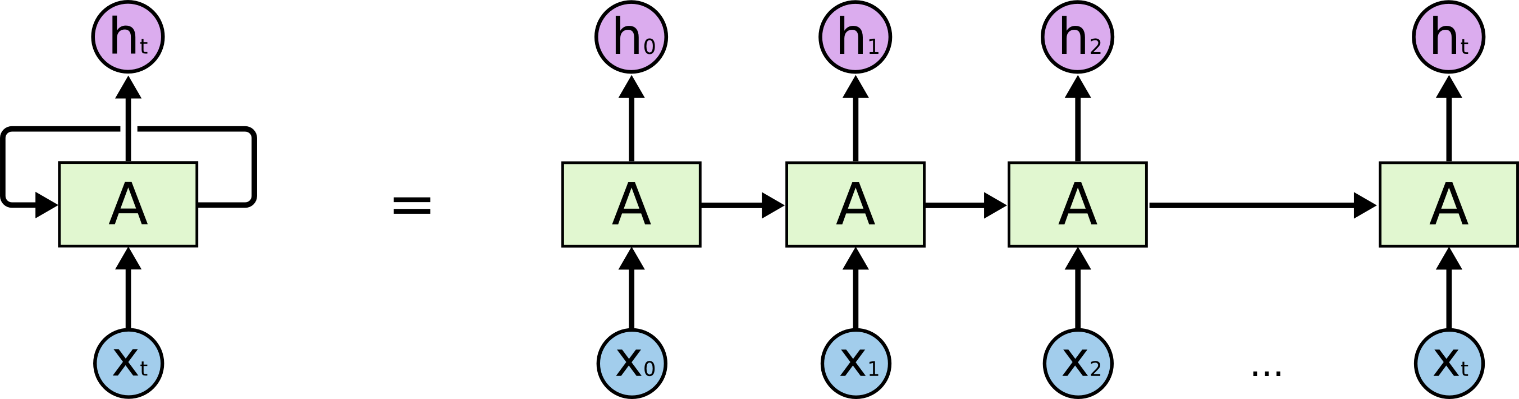


3.5 RNN

In the above flow chart, a piece of neural system, **A**, gets some input info **Xt** and yields a worth **ht**. A loop enables data to be passed starting with one stage of the system then onto the next.

These circles cause Recurrent neural systems to appear to be somewhat puzzling.

Nonetheless, on the off chance that you think more, things being what they are, they aren't too not quite the same as a typical neural system. A Recurrent neural system can be thought of as different duplicates of a similar system, each passing a message to a successor.



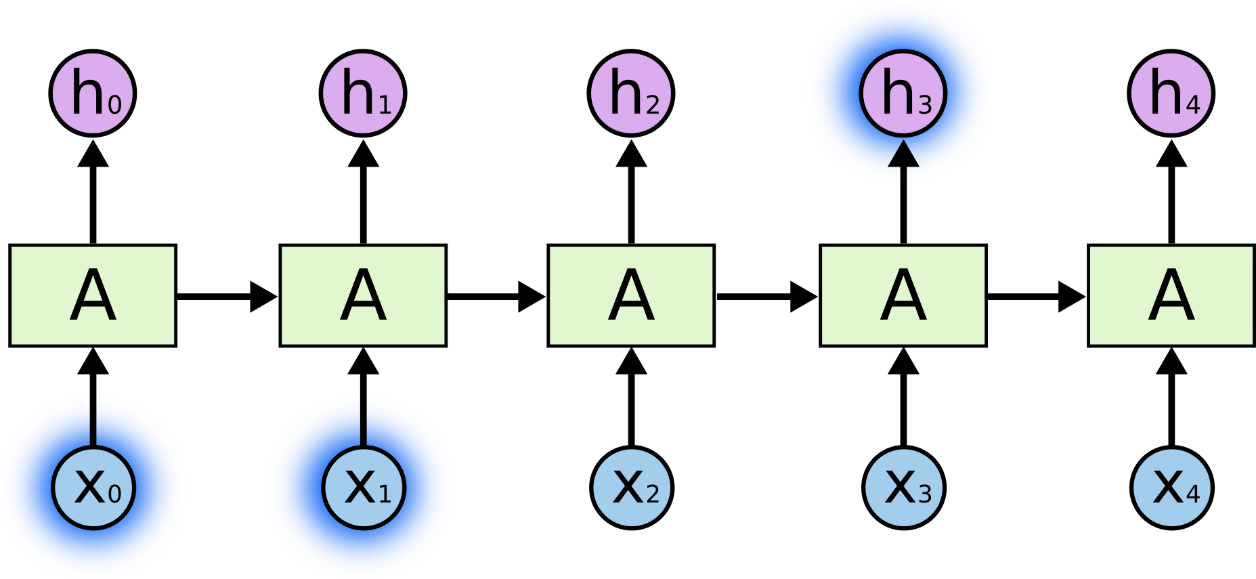
Think about what occurs on the off chance that we unroll the circle: This chain-like nature uncovers that Recurrent neural systems are personally identified with lists and sequences. They're the common design of neural system to use for such information. Over the most recent couple of years, there have been mined blowing achievement applying Recurrent neural network systems to an assortment of issues: image captioning, language modelling, speech recognition, interpretation, picture inscribing… The rundown goes on. Essential to these successes is the use of “LSTM” , a very special kind of recurrent neural network which works, for many more tasks, and much better than the standard version of RNN. Practically all exciting outcomes dependent on intermittent neural systems are accomplished with them.

**The Problem of Long-Term Dependencies**

One of the interests of Recurrent Neural Networks is the possibility that they may have the option to interface past data to the present errand

In the event that Recurrent Neural Networks could do this, they'd be incredibly helpful.

But Sometimes, we only need to look at recent information to perform the present task. For instance, consider a language model attempting to anticipate the following word dependent on the past ones. In the event that we are attempting to foresee the final say regarding "the mists are in the sky," we needn't bother with any further setting – it's really evident the following word



In such cases, where the hole between the pertinent data and the spot that it's required is little, Recurrent Neural Networks can figure out how to utilize the past data.

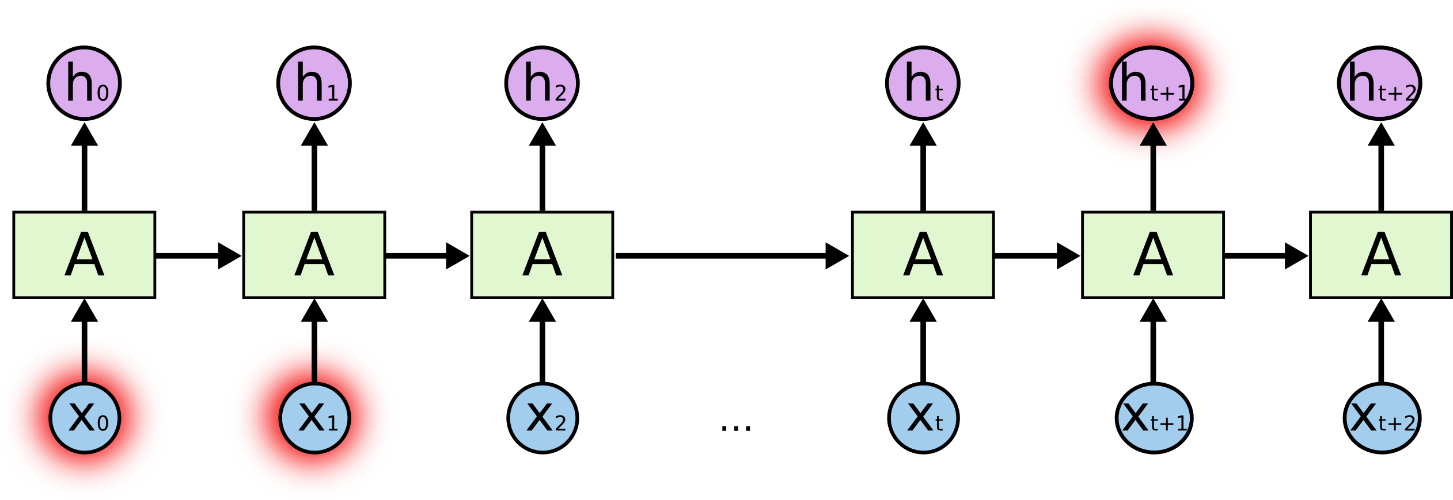
Be that as it may, there are likewise situations where we need more setting.

Consider attempting to foresee the final say regarding the content "I experienced childhood in France… I talk familiar French." Recent data proposes that the following word is most likely the

name of a language, yet on the off chance that we need to limit which language, we need the setting of France, from further back.

It's completely feasible for the hole between the important data and where it is expected to turn out to be exceptionally enormous.

Lamentably, as that hole develops, Recurrent Neural Networks become unfit to figure out how to interface the data.



In principle, Recurrent Neural Networks are completely fit for taking care of such long-term dependencies A human could choose parameters for them to take care of toy issues of this structure.

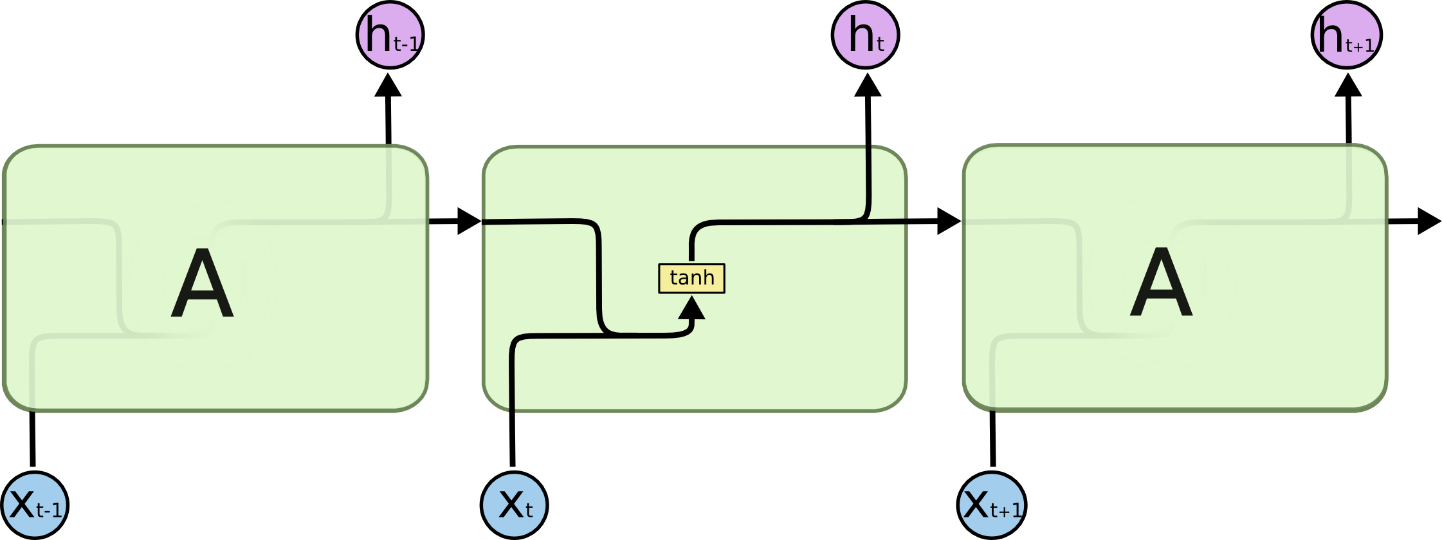
Sadly, in practice, Recurrent neural system doesn’t seem to be able to learn them.

Thankfully, Long Short Term Memory systems don’t have this problem!

**Long Short Term Memory Networks**

Long Short Term Memory networks – usually just called LSTM – are a special kind of Recurrent Neural Networks that are capable of learning long term dependencies. Hochreiter & Schmidhuber, introduced then and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used.

Long Short Term Memories are explicitly designed to avoid the long-term dependency problem. Recollecting data for significant stretches of time is for all intents and purposes their default conduct, not something they battle to learn.



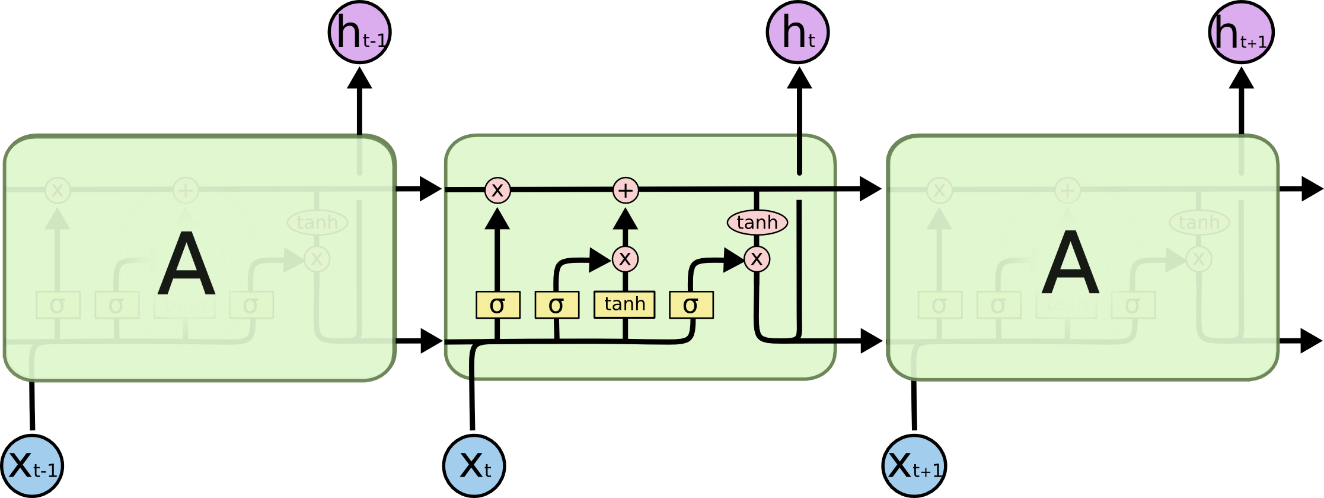
3.9 LSTM

All intermittent neural systems have the type of a chain of rehashing modules of neural system.

In standard Recurrent Neural Networks, this rehashing module will have a basic structure, for example, a solitary tanh layer.

The rehashing module in a standard Recurrent Neural Networks contains a solitary layer.

Long Short Term Memory's also have this chain like structure, but the repeating module has a different structure. Rather than having a solitary neural system layer, there are four, interfacing in an extraordinary way.



The repeating module in a Long Short Term Memory contains four interacting layers.

For the time being, simply attempt to get settled with the documentation we'll be utilizing.

In the above flow chart, each line conveys a whole vector, from the yield of one hub to the contributions of others. The pink circles represent point wise operations, like vector expansion, while the yellow boxes are already trained neural system layers.

Lines blending signify link, while a line forking indicate its substance being replicated and the duplicates going to various areas.

**The Core Idea Behind Long Short Term Memory**

The key to Long Short Term Memory's is the cell state, the horizontal line running through the top of the diagram.

The cell state is somewhat similar to a transferring line. It runs straight down the whole chain, with just some minor direct cooperation. It's simple for data to simply stream along it unaltered.



The Long Short Term Memory has the capacity to evacuate or add data to the phone state, deliberately directed by structures called doors. Entryways are an alternate way to let data through. They are made out of a sigmoid neural net layer and a point wise increase activity.



The sigmoid layer yields numbers somewhere in the range of zero and one, portraying the amount of every segment ought to be let through.

An estimation of zero signifies "let nothing through," while an estimation of one signifies "let everything through!

A Long Short Term Memory has three of these gates, to protect and control the cell state.

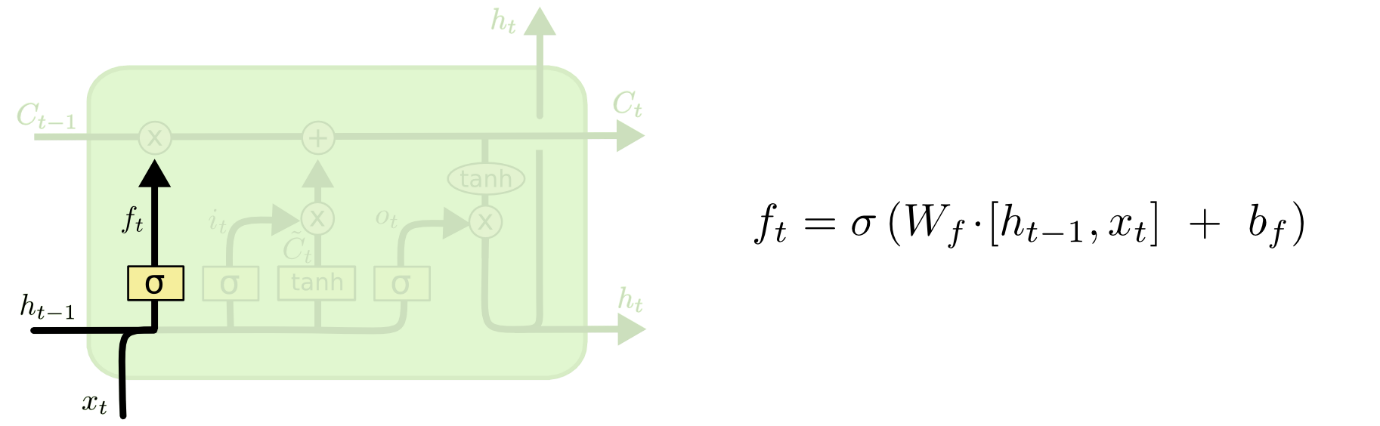
**Step-by-Step Long Short Term Memory Walk Through**

The first step in our Long Short Term Memory is to decide what information we’re going to throw away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It looks at ht−1 and XT, and outputs a number between 0 and 1 for each number in the cell state Ct−1.

A 1 speaks to totally keep this while a 0 speaks to totally dispose of this

We should return to our case of a language model attempting to anticipate the following word dependent on all the past ones. In such an issue, the cell state may incorporate the sexual orientation of the present subject, with the goal that the right pronouns can be utilized.

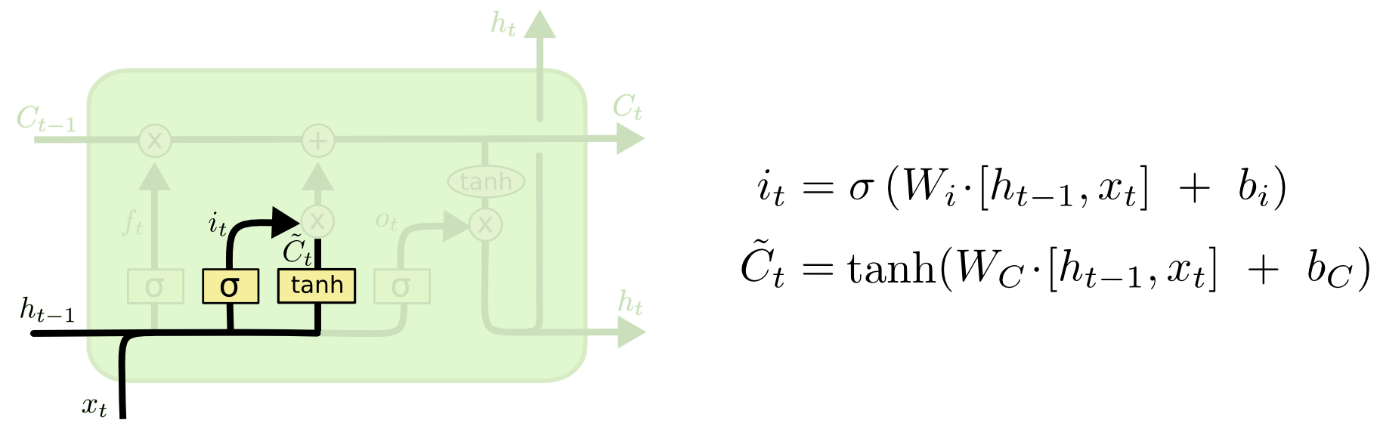
At the point when we see another subject, we need to overlook the sexual orientation of the old subject.



The following stage is to choose what new data we're going to store in the cell state.

Initial, a sigmoid layer called the "input entryway layer" chooses which esteems we'll refresh.

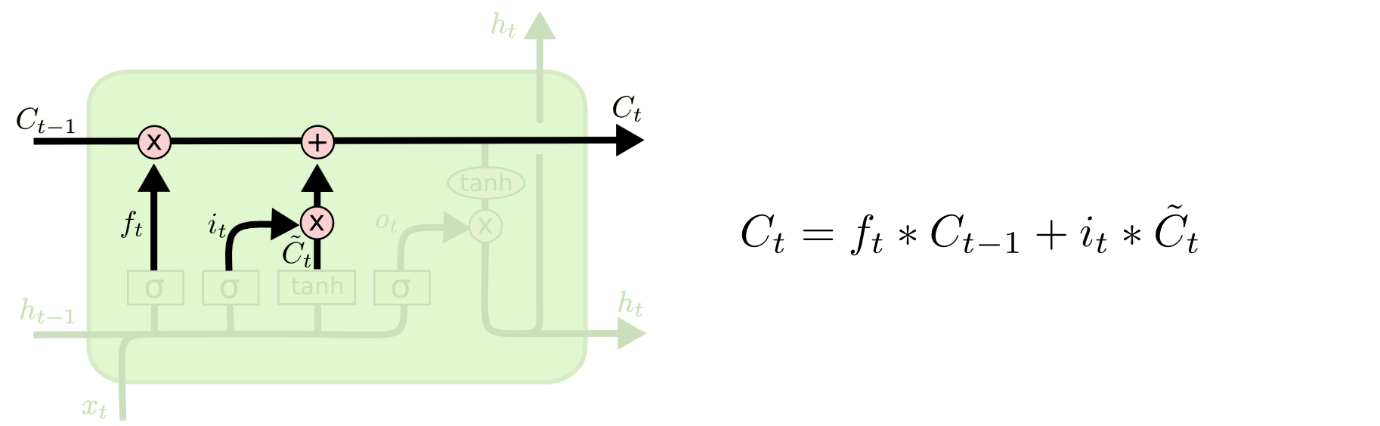
Next, a tanh layer makes a vector of new competitor values, C~t, that could be added to the state.



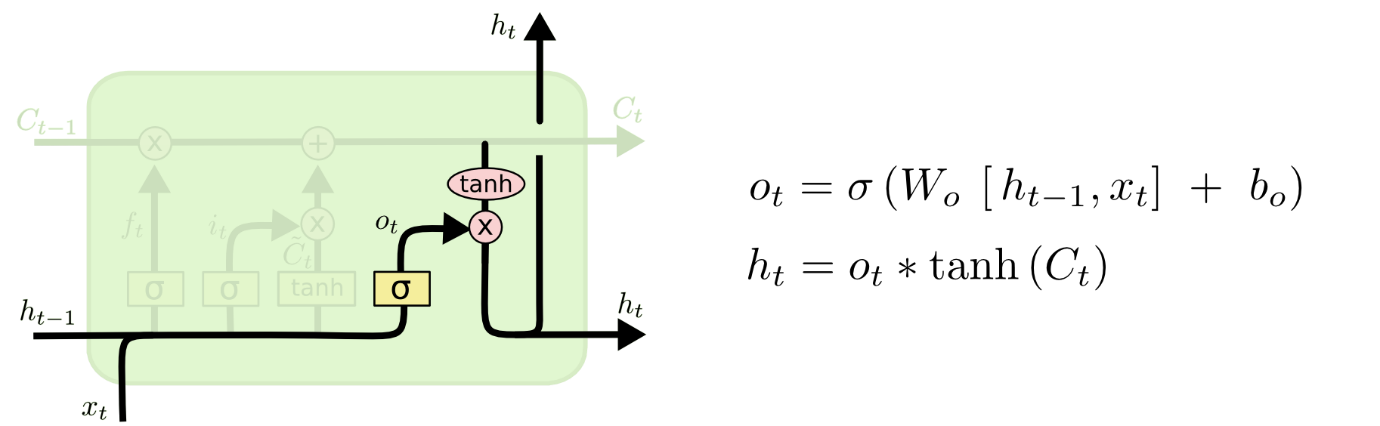
In the subsequent stage, we'll consolidate these two to make an update to the state.

In the case of our language model, we'd need to include the sexual orientation of the new subject to the cell state, to supplant the bygone one we're overlooking. It's presently time to refresh the old cell state, Ct−1, into the new cell state Ct. The past advances previously chose what to do, we simply need to really do it.

We increase the old state by ft., overlooking the things we chose to overlook before.



This is the new up-and-comer esteems, scaled by the amount we chose to refresh each state esteem. On account of the language model, this is the place we'd really drop the data about the old subject's sexual orientation and include the new data, as we chose in the at long last, we have to choose what we're going to yield. This yield will be founded on our cell state, however will be a sifted adaptation. To begin with, we run a sigmoid layer which chooses what parts of the cell state we're going to yield.



At that point, we put the cell state through tanh (to push the qualities to be somewhere in the range of −1 and 1) and duplicate it by the yield of the sigmoid entryway, so that we just yield the parts we chose to. For the language model, since it just observed a subject, it should yield data significant to an action word, on the off chance that that is what is coming straightaway.

**Bidirectional LSTMs**

The possibility of Bidirectional Recurrent Neural Networks (RNNs) is direct. It includes copying the principal intermittent layer in the network so that there are currently two layers' one next to the other, at that point giving the input sequence as is as contribution to the primary layer also, giving a switched duplicate of the input sequences to the second.

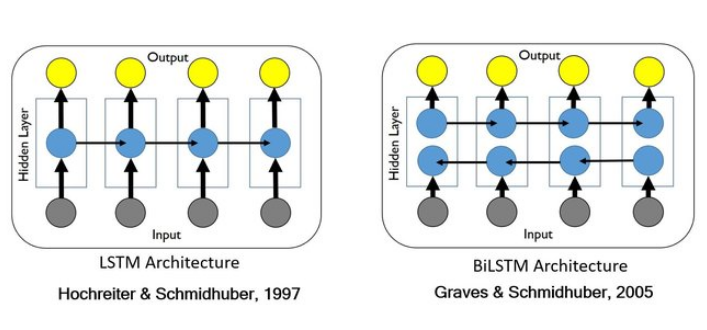


Fig 3.16 BI-LSTM

To beat the confinements of a customary Recurrent Neural Networks, we propose a bidirectional recurrent neural system (BRNN) that can be prepared utilizing all accessible info data previously.

The thought is to part the state neurons of a customary Recurrent Neural Networks in a section that is liable for the positive time course (forward states) and a section for the negative time direction (backward states)

This approach has been used to great effect with Long Short-Term Memory (LSTM) Recurrent Neural Networks.

The use of providing the sequence bi-direction ally was initially justified in the domain of speech recognition because there is evidence that the context of the whole utterance is used to translate what is being said as opposed to a straight understanding.

… relying on knowledge of the future seems at first sight to violate causality. Sounds, words, and even whole sentences that at first mean nothing are found to make sense in the light of future context. What we should recall is the differentiation between errands that are genuinely on the web – requiring a result after each input – and those where yields are just required at only needed at the end of some input segment.

The use of bidirectional LSTMs may not make sense for all sequence prediction problems, but can offer some benefit in terms of better results to those domains where it is appropriate.

We have found that bidirectional networks are significantly more effective than unidirectional ones…

To be clear, time steps in the input sequence are still processed one at a time, it is just the network steps through the input sequence in both directions at the same time.

Bidirectional LSTMs in Keras

keras is an open source profound learning library written in python.it is fit for running on the highest point of MXNET, Deeplearning4j, Tensorflow, CNTK or Theano, deeplearning4j, tensorflow or Theano. in 2017, Google’s tensorflow group chose to help keras in tensor flow’s centre library.

Bidirectional Long Short-Term Memories are supported in Keras via the Bidirectional layer wrapper.

This wrapper takes a recurrent layer (e.g. the first LSTM layer) as an argument.

It also allows you to specify the merge mode, that is how the forward and backward outputs should be combined before being passed on to the next layer. The options are:

• ‘sum ‘: With the help of this the outputs are added together.

• ‘mul ‘: With the help of this outputs are multiplied together.

• ‘concat ‘: With the help of this outputs are concatenated together, providing double the number of outputs to the next layer.

• ‘ave ‘: With the help of this average of the outputs is taken.

The default mode is to concatenate, and this is the method often used in studies of bidirectional Long Short-Term Memory.

**TensorFlow** is the chief open-source Deep learning system created and kept up by Google. Although using TensorFlow directly can be challenging, the modern tf.keras API brings the simplicity and ease of use of Keras to the TensorFlow project.

Utilizing tf.keras enables you to design, configuration, fit, assess, and utilize profound deep learning models to make predictions and analysis in only a small of lines of code. It makes common deep learning tasks, such as classification and regression predictive modelling, accessible to average developers looking to get things done.

**Keras** is an open-source library written in Python for applying deep learning algorithms on data.

The project was started in 2015 by Francois Chollet. It quickly became a popular framework for developers, becoming one of, if not the most, popular deep learning libraries.

During the period of 2015-2019, developing deep learning models using mathematical libraries like TensorFlow, Theano, and PyTorch was cumbersome, requiring tens or even hundreds of lines of code to achieve the simplest tasks. The focal point of these libraries was on research, adaptability, and speed, not convenience.

Keras was popular because the API was clean and simple, allowing standard deep learning models to be defined, fit, and evaluated in just a few lines of code.

A secondary reason Keras took-off was because it allowed you to use any one among the range of popular deep learning mathematical libraries as the backend (e.g. used to perform the computation), such as TensorFlow, Theano, and later, CNTK. This allowed the power of these libraries to be harnessed (e.g. GPUs) with a very clean and simple interface.

In 2019, Google released a new version of their TensorFlow deep learning library (TensorFlow 2) that integrated the Keras API directly and promoted this interface as the default or standard interface for deep learning development on the platform.

This integration is commonly referred to as the tf. keras interface or API (“tf” is short for “TensorFlow“). This is to distinguish it from the so-called standalone Keras open source project.

• Standalone Keras. The standalone open source project that supports TensorFlow, Theano and CNTK back ends.

• tf.keras. The Keras API integrated into TensorFlow 2.

The Keras API implementation in Keras is referred to as “tf.keras” because this is the Python idiom used when referencing the API. First, the TensorFlow module is imported and named “tf “; then, Keras API elements are accessed via calls to tf. keras. Given that TensorFlow was the de facto standard backend for the Keras open source project, the integration means that a single library can now be used instead of two separate libraries. Further, the standalone Keras project now recommends all future Keras development use the tf.keras API.

**V. Evaluation**

At this stage the model (or models) obtained are more thoroughly evaluated, assessment of data mining techniques with respect to business success criteria, the assessment of the model used and the steps executed to build the model are assessed to be sure it appropriately accomplishes the business targets. Complete review is done of the process and lists are made of the possible actions and outcomes of the project. In this research we created to models which we will be evaluating based on the results we obtained for our sentimental classification process. We created our baseline deeplearning model using rapidminer while we used Jupyter notebook and tensorflow framework to implement Bi-LSTM layer Model. We will be discussing more about the results obtained on both models and their evaluation and conclusions from this research in the next chapters but were able to get desired results with the help of this technique of using CRISP-DM Model for our analysis.

**VI. Deployment**

Creating the model is generally not the end of the project. Even if the purpose of the model is to increase the knowledge of the data, the knowledge gained from it will need to be organized and presented in a way to the customer which can be utilized. At last deployment is planned, with a proper maintenance and monitoring. The final report is made and there is usually a presentation to discuss its finding and at last the project review is done. Once the models are created, the knowledge gained is presented in a way that the Business can use it, this includes using the models created to aid the decision making process of the organization. Based on the requirement, the deployment phase can be a simple generated report or a repeated process. With the help of this research, the deployment phase will be implementing the LSTM Model for sentimental analysis on the new data recorded on TripAdvisor site so as to classify the reviews of new costumers into positive or negative. This can also be used on other different Arabic dialect websites to classify the sentiments of different users about different products.

**Chapter 4 Analysis/Discussion**

This chapter is divided into two parts, which we will be looking into in detail one by one

4.1 Exploratory Data Analysis

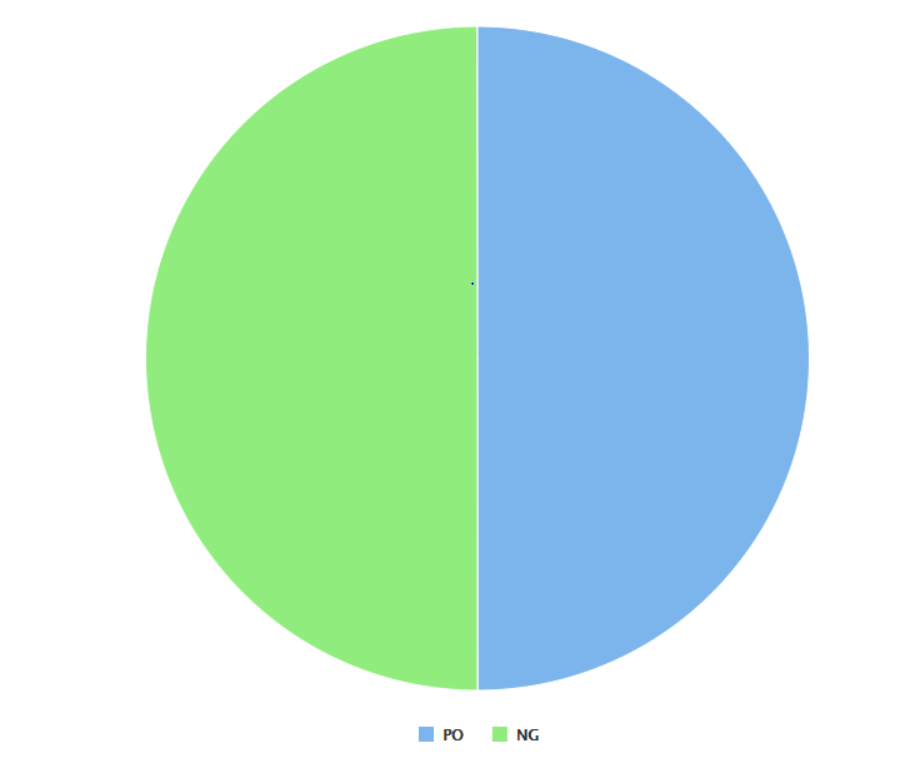
4.2 Model results:

**4.1 Exploratory Data Analysis**

Exploratory data analysis (EDA) is conducted on our dataset before the models are applied on it, to analyse and identify the main characteristics of the data through visualizations before the models are applied on it



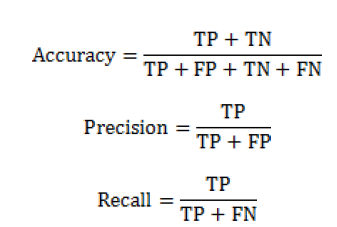
The above word cloud represents the amount of positive to negative polarities ratio within the dataset, as it is clear that the amount of positives is huge compared to negative ones hence we will be making them equal before applying our model to the data so that we have equal amount of examples for classification training for our sentimental analysis.



The above pie is the count of polarities that are now both equal to one another after we applied smote upsmapling to balance the dataset so that model can be trained to predict both the polarities of sentiments among people correctly.

**4.2 Model results**

In this section we will be discussing the results of our research by analysing the results we obtain from both the models and then comparing them. We will be comparing the results of our baseline model, where we applied machine learning models on our datasets which performed well giving good results for few models like fast large margin, generalised linear model, logistic regression and deep learning and then we will compare these results with the results we obtained after implementing Bi-LSTM Model in tensorflow. We will be deciding based on accuracy to choose which model.



Where

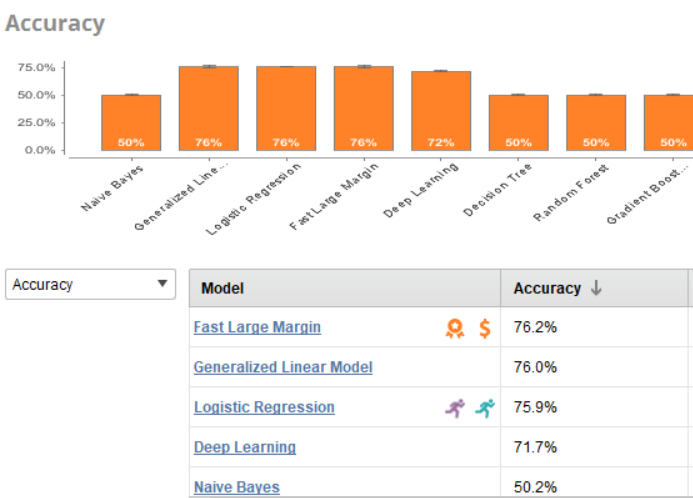
TP= True positive,

TN=True negative,

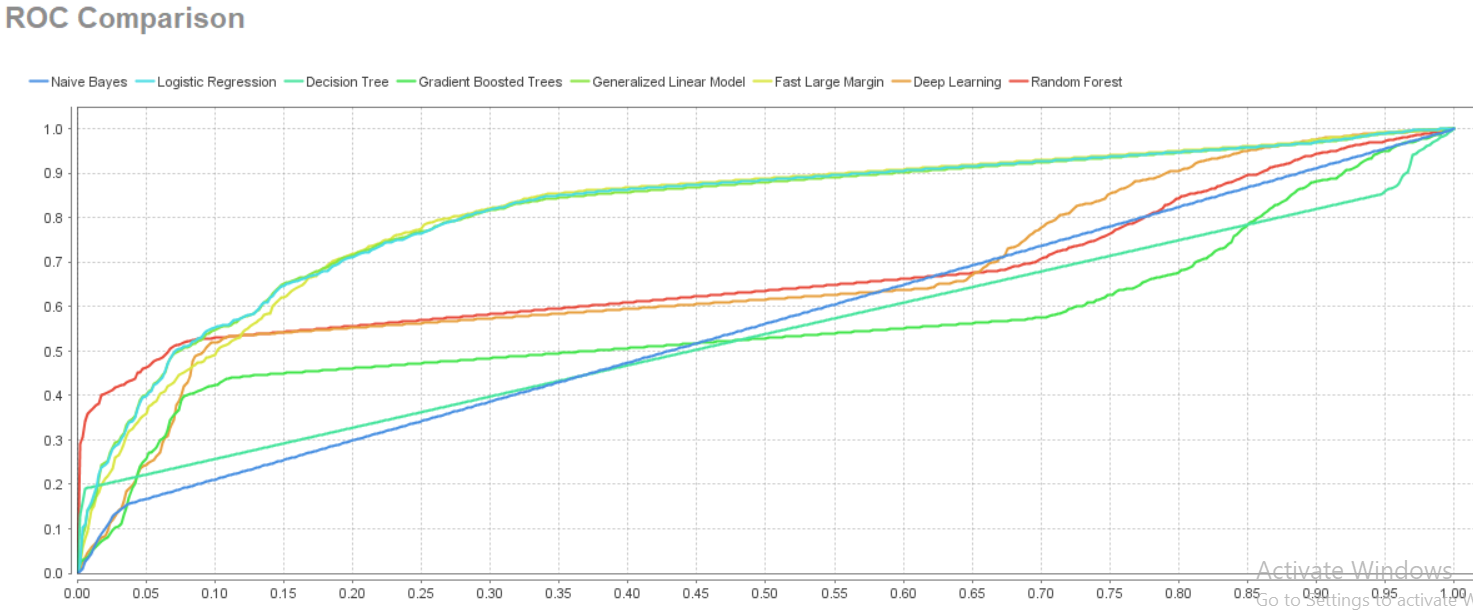
FP=False positive,

FN=False negative.

In our base model where we implemented multiple machine learning algorithms to see which one is performing best for this type of supervised sentimental classification, the best one came with highest accuracy is fast large margin with 76.2% accuracy, with generalised linear model (76%) , logistic regression (75.9%) and deeplearning (71.9%) following it.



**ROC COMPARISON**

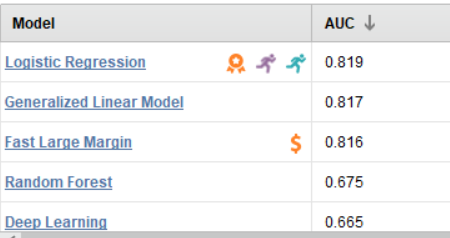


This plot tells you a few different things.

A model that predicts at chance will have an Receiver Operating Characteristic curve that looks like the diagonal line. That is not a discriminating model.

The further the curve is from the diagonal line, the better the model is at discriminating between positives and negatives in general.

There are useful statistics that can be calculated from this curve, like the Area Under the Curve (AUC) and the Youden Index.



After implementing the same dataset to our new model with bi-LSTM layer, the results we obtained were considerably way better than the machine learning models. For training set we got the accuracy of 96% While with test set we got the accuracy of 93%. When we do a blind test on this, by choosing 400 random samples of data, the number of reviews it was successfully able to distinguish was 90% i.e. it was able to correctly predict 362 out of 400.

For LSTM and Bi-LSTM, we use the following configuration:

1. Number of LSTM units = 100, Recurrent

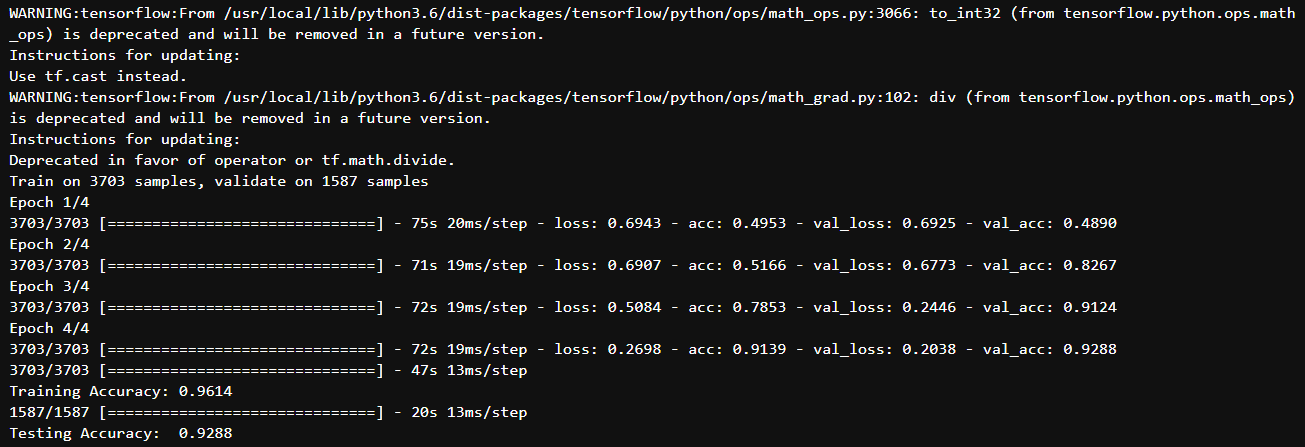
dropout = 0.2

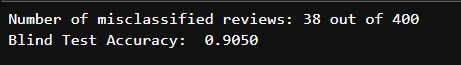
2. Loss function: Binary cross-entropy loss

3. Recurrent Activation: Hard sigmoid

4. Activation: tanh

We report Precision, Recall, F-score and accuracy values using methods in scikitlearn





As it is clear from the results deeplearning models with LSTM Layer, in this case Bi-LSTM Layer performed significantly well compared to machine learning algorithm models.

**Challenges Faced**

Challenges of colloquial Arabic language:

1.Use of dialects- the presence of the dialects in the Arabic writings have made the sentiment analysis task additionally challenging for testing, because of the nonappearance of explicit standards that oversee the composition or talking framework.

in Arabic language there are 6 dominant dialects which are further divided into more complex dialects

some dialects lacks spelling standards compared to MSA

2.handing compound phrases and idioms that are natural for an Arabic speaker are sometimes difficult to understand for a machine based text classification models and hence can sometimes be misclassified and decrease the performance of the model

3.subjectivity and sentiment analysis challenge

• the need for entity name recognition-names are often adjectives in Arabic.

• 2. handling negation

4.Reviews specific challenges

• Mix of dialects

• Transliterated words

• Informality of the language

• Contradictory language

• Sarcasm content

• Unclear content

5. In contrast to different dialects, the Arabic language has a morphological multifaceted nature which makes the Arabic slant examination a difficult undertaking.

Moreover, Generally, one of the problems of sentiment analysis is the high dimensionality of the feature vector.most resources developed within studies addressing Arabic sentiment analysis, are either limited in size, not publicly available or developed for a very specific domain

**Chapter 5 – Conclusion**

In this study, we introduced large datasets for sentiment analysis, scrapped from TripAdvisor websites that support reviews in the domains of hotels. The dataset contained text in Arabic dialect and user’s sentiment polarities i.e. positive review or negative review.

Although the generated lexicon isn’t very large, experimental results have shown that abstracting reviews by lexicon based features only, achieved a relatively fair performance for the task of sentiment classification.

An extensive set of experiments was performed for the sake of benchmarking the datasets and testing their viability for two class sentiment classification problems. Out of the experimental results, we highlighted that the top performing classifier was FAST LARGE MARGIN and the worst was Decision tree, and that the best performing feature representations were the blend of the vocabulary based features with different features.

Bi-LSTM or bi-directional long short term memory network models works way better than any other machine learning models, performing significantly well the with test with good accuracy and fast implementation for both trained and testing data as well as blind data test.

LSTM models are very effective for sentimental classification analysis, and can be useful in many other applications where text based classifications are required.

Finally, according to the error analysis on the task of sentiment classification, we find that the document length and richness with subjectivity both affect the accuracy of sentiment classification, in which; sentiment classifiers tend to work better when the documents are rich with polar terms of one class, i.e., high values of subjectivity score. However, this often doesn’t hold when the document length is extremely short or long.

Although the generated datasets cover multiple domains, they are all generated only from reviews. Thus, their usefulness for social media sentiment analysis, is yet to be studied. This might include generation of additional datasets to cover cases that doesn’t show up in the reviews domain but common in social media like advertisements and news. This is a motivation for future research work.

**Future Work**

The future work of this research will be extending the number of algorithms used and creating a sentimental analysis tool for an Organization that can make good use of it.

Using this research as base, we can exploit other LSTM Models as well. LSTM are very well performing in regards to text classification and related applications and there are many other LSTM Models that are being studied and can be applied to get better results.

The findings from this research can be used as a foundation for building a sentimental analysing tool which can be used for other languages also, not just Arabic or English but many more distinct and less explored, and also a tool that not just analyse one language at a time but doing sentimental analysis on a dataset that may contain data from multiple dialects.

We can also work on code-mixed data where data contains one language that is combined with to other to express their sentiments in a more efficient way like twitter users using Hindi language but typing it in English because of ease of typing and available qwerty keypads.

Something that we might want to attempt later on is the utilization of word

embeddings explicitly prepared for the SA task, just as considerably increasingly complex Deep Learning models, for instance those that utilization a consideration system. Although the generated datasets cover multiple domains, they are all generated only from reviews. Thus, their usefulness for social media sentiment analysis, is yet to be studied. This might include generation of additional datasets to cover cases that doesn’t show up in the reviews domain but common in social media like advertisements and news. This is a motivation for future research work.

## Chapter 7 – Plagiarism and Referencing

1. Abdul-mageed, M., Diab, M. (2012) : “AWATIF: A Multi-Genre Corpus for Modern Standard Arabic Subjectivity and Sentiment Analysis”. Lrec. 3907–3914.
2. Abdul-mageed, M., Diab, M. (2014): “SANA: A Large Scale Multi-Genre, Multi-Dialect Lexicon for Arabic Subjectivity and Sentiment Analysis”. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14). pp. 1162–1169.
3. Abdul-Mageed, M., Diab, M (2012).: “Toward Building a Large-Scale Arabic Sentiment Lexicon”. Proceedings of the 6th International Global WordNet Conference. pp. 18–22.
4. Aly, M., Atiya, A. (2013): “LABR: A Large Scale Arabic Book Reviews Dataset”. Aclweb.Org. 494–498.
5. El-Beltagy, S., Ali, A (2013): “Open Issues in the Sentiment Analysis of Arabic Social Media: A Case Study. in proceedings of 9th International Conference on Innovations in Information Technology (IIT)”. pp. 215 – 220.
6. Elsahar, H., El-Beltagy, S. (2014): A Fully Automated Approach for Arabic Slang Lexicon Extraction from Microblogs. Computational Linguistics and Intelligent Text Processing. pp. 79–91 Springer Berlin Heidelberg.
7. Maamouri, M. et al. (2004).: “The penn Arabic treebank: Building a large-scale annotated Arabic corpus. NEMLAR Conference on Arabic Language Resources and Tools”. pp. 102–109.
8. Nabil, M. et al. (2014).: “LABR: A Large Scale Arabic Book Reviews Dataset. arXiv Prepr. arXiv1411.6718”.
9. Rushdi-Saleh, M., Martin-Valdivia, T. (2011): “OCA: Opinion corpus for Arabic. J. Am. Soc. Inf. Sci. Technol. 62.10. 2045–2054”.
10. Rehab M Duwairi, Raed Marji, Narmeen Sha'ban, and Sally Rushaidat (2014):” Sentiment analysis in Arabic tweets. In 2014 5th International Conference on Information and Communication Systems (ICICS), pages 1{6. IEEE.
11. Maha Heikal, Marwan Torki, and Nagwa El-Makky (2018). “Sentiment analysis of Arabic

tweets using deep learning”. Procedia Computer Science, 142:114{122.

1. Muhammad Abdul-Mageed, Mona Diab, and Sandra Kubler. Samar (2014): Subjectivity and sentiment analysis for Arabic social media. Computer Speech & Language,28(1):20{37,
2. Ahmad Al Sallab, Hazem Hajj, Gilbert Badaro, Ramy Baly, Wassim El Hajj, and Khaled Bashir Shaban (2015). Deep learning models for sentiment analysis in Arabic. In Proceedings of the second workshop on Arabic natural language processing, pages

9-17.

1. Ramy Baly, Georges El-Khoury, Rawan Moukalled, Rita Aoun, Hazem Hajj, Khaled Bashir Shaban, and Wassim El-Hajj (2017). “Comparative evaluation of sentiment

analysis methods across Arabic dialects”. Procedia Computer Science, 117:266{273,

.

1. Assia Soumeur, Mheni Mokdadi, Ahmed Guessoum, and Amina Daoud (2018). “Sentiment analysis of users on social networks: Overcoming the challenge of the loose usages of the Algerian dialect”. Procedia computer science, 142:26{37.
2. Malak Abdullah, Mirsad Hadzikadicy, and Samira Shaikhz. Sedat (2018): “Sentiment and emotion detection in Arabic text using cnn-lstm deep learning”. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pages

835{840. IEEE.

1. Abbasi, A., H. Chen, and A. Salem (2008), “Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums”. ACM Trans. Inform. Syst,. 26(3): p. 12.
2. Duwairi, R. and M. El-Orfali (2014), “A study of the effects of pre-processing strategies on sentiment analysis for Arabic text”. Journal of Information Science, 40(4): p. 501-513.
3. Shoukry, A. and A. Rafea (2015.), “A hybrid approach for sentiment classification of Egyptian Dialect Tweets”. in 2015 First International Conference on Arabic Computational Linguistics (ACLing). IEEE.
4. Khalil, T., et al. (2015), “Which configuration works best? an experimental study on supervised Arabic twitter sentiment analysis”. in 2015 First International Conference on Arabic Computational Linguistics (ACLing). IEEE.
5. Aliane, A., et al. (2016.), “A genetic algorithm feature selection based approach for Arabic Sentiment Classification”. in IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). IEEE.
6. El-Naggar, N., Y. El-Sonbaty, and M.A. El-Nasr (2017), “Sentiment analysis of modern standard Arabic and Egyptian Dialectal Arabic tweets”. in 2017 Computing Conference. IEEE.
7. Mourad, A. and K. Darwish (. 2013), “Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs”. in Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis.
8. El-Makky, N., et al. (2014), “Sentiment analysis of colloquial Arabic tweets”. in ASE Big Data/Social Informatics/PASSAT/BioMedCom 2014 Conference, Harvard University.
9. Omar, N., et al., (2014) “A comparative study of feature selection and machine learning algorithms for Arabic sentiment classification”. in Asia information retrieval symposium. Springer.

## Appendices

This document will guide you through the contents of the Artefacts and the necessary steps to implement the python code for dissertation project titled “Arabic Sentimental Analysis with Deep learning LSTM”.

### I. Contents of the Artefacts

#### **1. Datasets**

* **HTL\_Transformed.csv:** The pre-processed data used in rapidminer process
* **HTL:** The pre-processed data for python code.

#### **2. Model**

* **Final result**

Rapidminer process results of all the machine learning models

* **Rapidminer Process for SA.rmp – Ra**pid miner process for machine learning models implementation

#### **3. python code**

**sentiment\_analysis\_arabic\_hotel\_reviews. ipynb:** Primary code required to execute sentimental analysis Bi-LSTM model.

**4. Readme:**  Explains the contents of the Artefacts, how to implement the code on jupyter.

**Please refer to the readme file attached along with the Artefacts to understand how the code is implemented in R studio and the installation process of H2O Driverless AI on Azure cloud.**