Abstract

Neural network technologies have been proven to be a game changer for many domains. One of its widely known applications is sales forecasting. This article presents a new approach which focuses on sales predictions using deep learning to give accurate forecast sales on the basis of sentiment analysis, by analyzing texts from Twitter that considers human behavior patterns and human decision, to extract the buying tendencies. Its originality is reflected in the fact of considering the sentiments not from the point of view of text tweets author, but their potential readers. The article discusses a practical application of this approach for car sales forecasts and specifically Tesla Model S.. The purpose of the work presented here is to give the foundation of a method called Smart Sales Forecast Engine by combining a sentiment analysis predicted by a Deep Recurrent Neural Networks (DRNN) model with historical sales of a specific product, like Tesla Model S, by using a Sequence to Sequence Recurrent Neural Network.

KeyWords: Sales Forecast Engine, Sales Forceast, Sentiment analysis, Word Embedding's, Deep Learning, Deep Recurrent Neural Network, Convolutional Neural Network, Sequence to Sequence Recurrent Neural Network.

1 Introduction

Almost all industrial sectors are experiencing strong competition, which forces them to rethink the life cycle of their products and which encourages them, among other things, to forecast sales and thus to produce intelligently and recoup their investments in order to increase their positioning on their market. To do this, manufacturers are resorting more and more to prediction techniques to make better forecasts and make the right decisions. At present, the literature offers a myriad of articles, each based on different methods and techniques, particularly those derived from artificial intelligence.

In this perspective, many articles focus on the sentiment analysis related to the automotive market [1][2][3]. This analysis is mainly done by analysing written texts on social networks, and social media. However, the majority of published articles essentially carry their judgment (or evaluation) on the brands that manufacture cars with the consideration of the authors sentiment of the studied texts. Their approach is useful in predicting the change in the financial valuation of a stock market, but do not give a good prediction for sales forecasts[1].

The article focuses on sales forecasts by following an original approach that goes against what is found in the literature. This takes into consideration the reader-oriented sentiment analysis of a text and not that of its writer. The illustration of this approach is done on the analysis of data collected on Twitter[2] between 2012 and 2017.

The research which is the subject of this article is interested in forecasts of sales in the automotive sector and focuses particularly on those of the TESLA brand and its models Tesla Model S. Such a choice is justified by the availability of data because of the great media coverage of this car model. Similarly, the interest in the automotive sector comes from the fact that it rarely encounters crises [8], its market is robust and highly competitive and it is one of the most competitive sectors. The automotive sector is currently experiencing considerable growth and is seeing new players come in and particularly attracted investors [8], the interest in the use of clean energy and autonomous vehicles.

In the following, the article provides an overview of the Smart Sales Forecast Engine (SSFE) methods and briefly introduces one of its fundamental aspects as the two psychological principles for evaluating people sentiment which are The Mere Exposure Effect [9][10] and The illusory truth effect[11] [12]. Then it's giving a

[1]As an example, in 2013 the image of Apple was splashed by the scandal of child labor work forces in one of its manufacturing factory [4]. Although Apple has received a negative press, its stock market has declined slightly but its sales have not been affected, on the contrary it has experienced a sharp increase of 26% the quarter that followed this scandal [5].

[2]Twitter is a microblogging service that allows its users to view small texts of 140 characters called tweets. Twitter is one of the fastest growing social media [6], has more than 500 million users, and is one of the most influential platforms [7].

short introduction of Deep Learning Architectures as Deep Recurrent Neural Networks (DRNN) [13] and Convolutional Neural Network (CNN)[14]. Also it presents the procedure to apply SSFE methods and details, in section 3, its application for the sales prediction of Tesla Model S car models with a concise explanation of the results obtained. A comparative study of the four most powerful deep learning architectures is explored in order to obtain the appropriate one for the SSFE method, for building and training an in-depth learning model.The last section shows how to use the SSFE forecast sales engine, combining a sentiment analysis predicted by a DRNN model with historical sales of a specific product, like Tesla Model S, by using a Sequence to Sequence Recurrent Neural Network[3] [15]. The article ends with a conclusion and reflects some of the current limitations of SSFE method that will be the prospects for its future improvements.

2 The fundamentals of SSFE method

The method SSFE is based on 3 fundamentals axes:

- Some psychological principles related to human behavior and human decisio [9][10][11][12].
- The deep learning architectures for Natural Language Processing and Time Series Prediction [16] [17]).
- A procedure to show how to apply it, step by step.

2.1 Psychological Principles for Evaluating Human Emotions

SSFE relies on two psychological principles to assess sentiments of a given person.

- The Mere Exposure Effect [9] [10]: this principle indicates that people tend to prefer things, ideas and people with whom they are familiar. In other words, the more a person feels familiar with things, the more he reacts positively.
- People tend to believe that the information they recurrently hear about are truths, thus creating an unconscious type of prejudice. Although the awareness of this prejudice wouldn't prevents its effect, in fact studies have shown that, the knowledge of the illusory truth effect, does not protect from its effects, it is beyond human control [11].

These principles have been studied for a long time[4] , but due to the nature of current recommendation systems and social media[20][21][22], their impacts have become more noticeable [23].
Among several problems of sentiment analysis systems found in literature is the treatment of emotions as exclusive classes, i.e. they treat sentiment classification as object classification (a table cannot be a table and an airplane, but a tweet can invoke more than one overlapping emotion strongly correlated with multiple sentiments, the works of Wang et al[24] and Zhou et al[25] perfectly illustrate these types of emotion-sentiment correlations), according to recent and old studies [26].

2.2 Deep Learning Architectures

When it comes to deep learning, the method is based on RNN (Recurrent Neural Network) and CNN (Convolutional Neural Network):

[3]Sequence to Sequence RNN are variety of an RNN that their input and output are sequence

[4]There are more than 34,000 possible emotions [18] but a study has shown that their number is unlimited[19]. This comes from the fact that society is constantly changing and every change brings new emotions.

- the RNN is an artificial neural network [27] that feeds the output neurons (states) of its inner layers as a new input at each time step (propagation / backpropagation) [28]. The simplest way to think about recurrent neural networks is to unfold them over time, as illustrated in Fig1 RNNs are specialized in the processing of sequence values [29], and can be scaled up to larger sequences better than a deep neural network (DNN) or a convolutional neural network(CNN) [27]. It can also process series of different lengths [30] and allows to share the parameters over several time steps to reveal the temporal dependencies of the values of the processed sequences [13]. Another aspect of an RNN is characterized by the many neuron activation functions[5] which requires a specific treatment for each one of them [31]. In the study that is the subject of this article, the main neuron activation functions[6] were tested and settled on a Gated Recurrent Unit (GRU[7]) cell [35]. Staking multiple RNNs on top of each other creates a Deep Recurrent Neural Network (DRNN) [13]

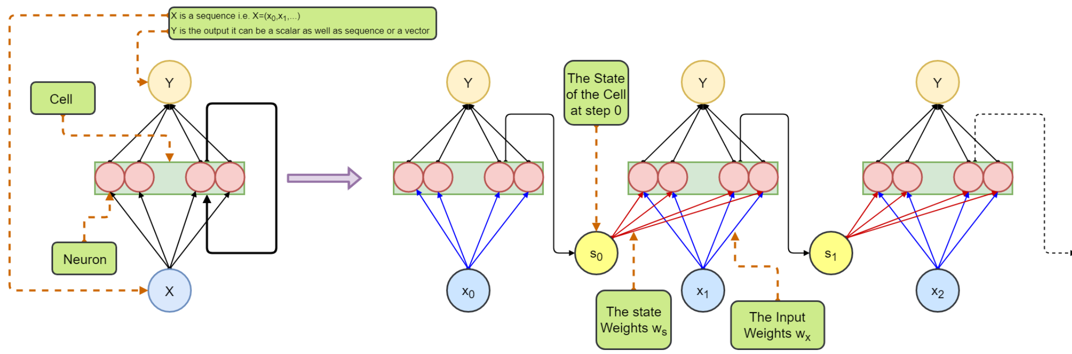


Figure 1: a sketch of recurrent neural network architecture

1. x_t is the input at time step t
2. s_t is the hidden stat of the neural network at time t, and it's calculated usually as a nonlinear combination of previous stat and new input $s_t = f(w_s s_{t-1} + w_x x_t)$, where f is a nonlinear activation function. So, a cell in a recurrent neural network has two sets of weights, ones for the input and others for the states.

- CNN currently characterizes the state-of-the-art summit in the detection of objects [36]. The simplest way to think CNN is that unlike Deep Neural Networks, which implement learnable weights, CNN uses learning filters to analyze data for hidden models.

  In this work, CNN is used in the comprative study (see sections 3.4.5) in order to determine which is the most appropriate Deep-Learning architecture for the SSFE method.

2.3 Procedure for applying the SSFE method

The proposed procedure, for the application of the SSFE method, is divided into four main parts which are detailed in section 3, and whose essential can be summarized as follows.

2.3.1 Collecting, filtering and selecting data

The purpose of this step is to collect from the Internet or databases (public or private, payable or not), a set of texts related to the context of the object of the forecasting study. These texts are then filtered to keep a good

[5]Activation neural networks are mathematical functions of which the choice is considered as a Hyper-Parameters in a neural network architecture.

[6]Any network of artificial neurons has a wide variety of activation functions [14].

[7]GRU cell was proposed by Kyunghyun Cho et al. [32] and was proven by Klaus Greff et al [33] to have the same performance as an Long Short Term Memory Cell[34].

selection.

2.3.2 Creation of a learning dataset

This step aims for labeling a large number of the previous selected texts taking into account the two of the main psychological principles of human decision mentioned above.

2.3.3 Word Embedding's

This step focuses on building an in-depth learning model and on training it to make it effective.
In order to choose the appropriate deep learning architecture among the four most powerful: RNN and CNN used for each two embedding approaches, one with Skip-Gram[8] and one with a neural network directly attached to their input serving as a layer of embedding.
Then, a comparison makes it possible to evaluate and classify them on the basis of their different rating criteria.

2.3.4 Forecasting Engine result

The goal of this step is to make the final sales forecast by combining sentiment analysis with historical sales of a specific product and by using a sequence to sequence Recurrent Neural Networks.

3 Application of the SSFE method

Before detailing the above-mentioned steps of the application procedure of the SSFE method, it is important to remember that this goes against what is done in sentiment analysis by interpretation of written texts. The approach usually encountered in the literature seeks to predict the author's sentiment [2], [37], [38]]by analyzing the context of the text he wrote in relation to his profile [39],[40]. SSFE goes beyond that because it is interested in predicting sales forecasts by analyzing the sentiments of tweets readers and in the impact that can have on their buying patterns.

3.1 Collecting, filtering and selecting data

This step focuses on collecting data from the Internet according to the context of the study to be conducted. The collection is done either by downloading an existing free or paid database (the most recommended way), either by the use of web-scarping tools (the most intelligent and particularly necessary in the case where no contextual database is available) or by progressive recovery from proprietary platforms or accounts.
In regard of data collection, for this article study, a repository on github made by "Jefferson-Henrique" [41] was used in order to collect several tweets over a period from January 2012 to December 2016. So, 10,000 tweets were studied in order to select the most recurring representatives as part of a set of sentiment data.

3.2 Data Labeling to create a learning dataset

As for the labeling process, it consists of attributing to each study text a label by analyzing its convergence with the two psychological principles mentioned above: the MereExposureEffect [9], [10] and the illusory truth effect [11], [12].

In the pilot application of SSFE exposed in this article, this step is to label the selected tweets to classify them according to their relevance to a car's brand or model. This is how the labeling process get 3 classes:

[8]The simple implementation of skip-gram model is train a neural network with a single hidden layer to do as follows: Given a word X find the most likely related Words $X_{i\geq 0}$ grammatically, semantically and contextually. The result of the Skip gram Model is an a embedings matrix which is represented by the network inner weights used for training. Mathematically the skip gram model is a sort of projection from a higher vector space with the size equals to the number of word in the corpus, into a lower vector with the size of the number of the inner hidden layer

neutral tweets that do not reflect any specific sentiment are labeled with a "0", tweets that reflect a negative sentiment are labeled with "-1" and tweets that reflect a positive sentiment are labeled with "1".

As illustrated in Figure 2, the labeling process begins with a tweet reading and its evaluation in relation to the context studied (brands and models cars) :

- The first level of evaluation makes it possible to assign a 0 label to neutral tweets or to go to the next level. The '0' means that the tweet do not reflect any tangible information about a car brand or without a possible purchase (see examples of lines1, 2, 3 of the table 1).
- The next level makes it possible to check if the text can affect a purchase trend. If so, go to the next step, otherwise a 0 label is assigned to the tweet (see example of lines 10, 22, 23, 24 of the table 1
- The last level allows attributing the label 1 if the tweet can positively influence a purchase (see examples of lines 4, 5, 6, 11, 12, 14, 15, 18 of the table 1) and -1 if its influence can be judged negative (see examples of lines 7, 8, 9, 16, 17, 21, 25 of the table 1

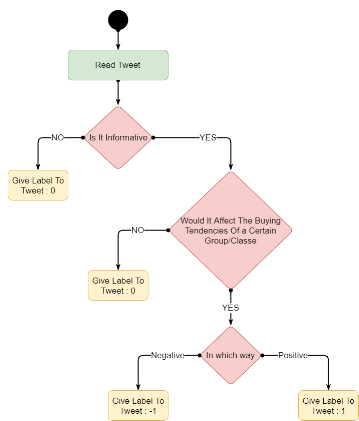


Figure 2: the labeling process of a tweet

3.3 Word Embedding's

Almost all Deep learning models that deals with natural language processing (NLP) [42]) need a numeric representation of text/document. This numeric representation can be as simple as the frequency of a given word like the traditional Bag Of Word approach[43] which can perform very well if the task is document classification (i.e. if a document is academic, a poem, a contract, ...) to more complex vector based representation.
Word representation is still one the most highly researched fields and thus many representation methodologies have been presented through the years, to cite a few:

- Glorotet et al [44] used a stacked Denoising Auto-encoder with sparse rectifier units that can perform "unsupervised text feature/representations" using labeled and unlabeled data.
- Zhang et al [45] used also an Auto-encoder architecture but in their research they used the model to learn task specific representations of textual data by relaxing the loss function from the label information.

[9]If person P "favors" product X made by brand A and new information comes as "brand B makes as good of a product Y like brand A makes product X" then person P will tend to "favor" product X over its competitors.

[10]Accusation are not addressed while being easy the disclaim i.e. Tesla has on board computer they could show the Logs

Label	Tweet Text	Reasoning
0	Fastest Production Cars In The World	Not informative
0	Shopping for cars in the rain	Not informative
0	my car's a wreck but so is my life so at least I'm consistent	Not informative
1	we purchased a Honda accord and after years it is still running like a champ, that's called getting your money's worth	Shows experiences of a customer
1	No other way to live without love a German car BMW do best	Shows enthusiast of a customer
1	Volvo Cars adds Microsoft's Skype for Business to its Series cars	Shows features of interest
-1	i think im honestly the dumbest person ever because i just chose a bmw	Shows regret of a purchase
-1	bMW i8 Smoke coming out from the vents interior	Shows defect of a product
-1	BMW recalls cars in China over airbag defects	Shows defect of a product
0	Tesla s Autopilot Isnt Afraid of Snow Covered Roads Breezes Through	Not informative enough to influence purchase
1	Tesla Autopilot Can Navigate On SnowCovered Road With No Lane Markings	New information uncovered affected positively the sentiment
1	Tesla model X reacted BEFORE the accident to avoid it	Shows capabilities of a product in reel testing
0	Are you test driving Self Driving Cars or maybe a Tesla	Not related
1	makes sense if tesla can make a better battery than anyone else similar to apple making arm processors	Shows brand strength by comparison to another popular brand[9]
1	German Environment Minister buys a Tesla Model S takes a swipe at domestic automakers in the	Shows Trust in the Brand by someone in power or a higher up
-1	Norwegian lawsuit says Tesla s horsepower claims are horses hhh well you get the picture	Shows legal ramification that could be issue due to public missleading
-1	Tesla Autopilot Update Imminent But Software Update Delayed	Shows product issues
1	Tesla Motors Inc Shines in Consumer Reports Survey	Shows experiences of a customer
0	Tesla Model S P85 Tesla Model S P85 Extended Warranty - Nearly Every Option - Excellent Condition	Not informative
0	South Korea may change EV incentive rule that hurts Tesla	Not efficient Information
-1	South Korea may change EV incentive rule that hurts Tesla	Not efficient Information
0	A case of he said	As More information

- Tang et al [46] used also a Neural Network based approach but in their work they used CNN and LSTM[11] to learn sentence representations, then a GRU to adaptively encode the semantic of sentences and their inherit relationship to the rest of the document.
- Dou [47] used LSTM for document representation and a Deep Memory Multi Layer Network for review ratings.
- AvoMuromägiet et al [48], used an ensemble word embedding's model that utilizes least squares regression and the solution to the orthogonal Procrustes problem.
- Kalchbrenneret et al [49] used a Dynamic CNN which with a dynamic k-max poling operator adding non linearity to the subsampling layers. Its feature graph is able to capture deep word inter-relationships.
- Socheret et al [50] used a Recursive Neural Tensor Network by using tensor based compositional function to improve on capturing the semantic, syntactic and contextual interactions within the document.
- And finally, one the commonly used word embedding algorithm Word2Vec which is, in its most basic form, without tweaks a Computationally-effective Neural Network model that learn vector representation of words using only the corpus/document without the need of a NLP-dictionary or any prerequisite. This algorithm comes with two types:
  - The Continuous Bag Of Word (CBOW) [51] with tries to predict a target word given a certain context.
  - The Skip-Gram Model [52] that tries to predict the context given a certain word.

In order to have the right way to apply the SSFE method by achieving an effective predictive result, two approaches were tested to word embeddings tweet's text. The first one is an extension of the above and uses the Skip-Gram model. The other one is to attach the embedding layers and let the model learn the weights based on the specified task. And as it pointed out in the presentation of the SSFE method, these two approaches, combined with the two deep-learning methods RNN and CNN, will allow us to test 4 architectures in order to compare their result and to choose the one whose learning is more reliable (see 3.4.5)

3.3.1 Word Embedding's using the Skip-Gram Model

The Skip-Gram Method applied here, is specifically the architecture introduced by Mikolovet al [52] because it's not computationally heavy, and has good performance by using Google TensorFlow [53].
The application of the Skip Gram model needs to go thought two main processes:

- Start by decomposing a corpus into words to build the training data set indicated by the algorithm 1. The goal here is to construct a dictionary with the couples of words forming the corpus[12] . As showing in the example of table 2, the algorithm 1 takes interest of the text word by word. In one iteration, it considers the current word (the bold one) and the words around-it using a step of 2 on its both sides (the underlined word). This rule does not apply if the current word is the first, the second, the penultimate or the last. So, for each iteration the algorithm is interested in a window whose size is 4 (the underlined word) and its skip is 1. This is more than enough considering that the dataset used is clean and regular.

[11]LSTM Long Short Term Memory cell is non linear mathematical function with multiple weights and biases that is proven to learn long term dependences in data[34]

[12]Words that are not in this dictionary are ignored and branded as 'UNK' token which means UNKNOW. For example, the following sentence 'Which boxerengined BMW is right for you We test them all backtoback to find out' will be transformed into 'Which UNK BMW is right for you We test them all UNK to find out'. The words 'boxerengined' and 'backtoback' are not frequent and they do not add any useful information about the tweet thus are disregard and replaced with the token 'UNK'. Now the new corpus words consist of 5000 words.

Source Text	Training Couple (Input,Output)
Tesla Motors Inc Shines in Consumer Reports Survey	(Tesla,Motors) (Tesla,Inc)
Tesla **Motors** Inc Shines in Consumer Reports Survey	(Motors,Tesla) (Motors,Inc) (Motors,Shines)
Tesla Motors **Inc** Shines in Consumer Reports Survey	(Inc,Tesla) (Inc,Motors) (Inc,Shines) (Inc,in)
Tesla Motors Inc **Shines** in Consumer Reports Survey	(Shines, Motors) (Shines,Inc) (Shines, in) (Shines, Consumer)
Tesla Motors Inc Shines **in** Consumer Reports Survey	(in,Inc) (in, Shines) (in, Consumer) (in, Reports)
Tesla Motors Inc Shines in **Consumer** Reports Survey	(Consumer, in) (Consumer, Shines) (Consumer, Reports) (Consumer, Survey)
Tesla Motors Inc Shines in Consumer **Reports** Survey	(Reports,in) (Reports, Consumer) (Reports, Survey)
Tesla Motors Inc Shines in Consumer Reports **Survey**	(Survey, Reports) (Survey, Consumer)

Table 2: example of creation of training data with couple words par decomposing a tweet in the corpus

Algorithm 1 Data preparation Algorithm

```
for each word in tweet get index ID  do
  if ID==0 then
    output='Words of index in the set 1,2'
  else if ID==1 then
    output='Words of index in the set 0,1,2'
  else if ID=='index the last word' then
    output='Words of index in the set ID-1,ID-2'
  else if  ID=='index of the pre last word' then
    output='Words of index in the set ID-1,ID-2,ID+1'
  else
    output='Words of index in the set ID-1,ID-2,ID+1,ID+2'
  end if
end for
```

- This step is to transform the (input, output) into a One-Hot encoded Vector [54].

The dictionary composed by the couples of words extract from the studied corpus, constitute the training data. These ones are simply plugging in the neural network and after 100000 training epochs, the neural weights are retrieved. Those weights act as an embedding matrix whose size is equal to "Number of words in the Corpus x Number of hidden layers" (5000 x 30 in the example). This embedding matrix characterizes the vector representation of all the words in the corpus.

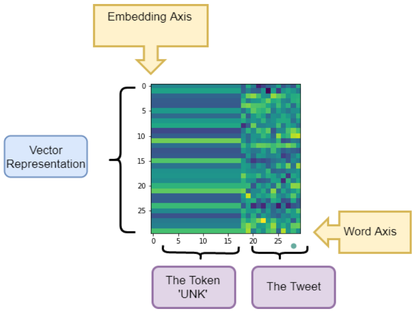
Applying this principle turns all tweets into matrices. Since tweets do not have the same length, in terms of the number of words, it is problematic to create these embedding matrices in order to use them in CNN / RNN. One solution to this problem is to pad all the tweets into affixed length which fixing, in the context study of this article, to 30 words (this value is logical because a tweet cannot exceed 140 characters). In consequence, for tweets whose word count is less than 30, the difference will be filled by adding one or more token "UNK" at the beginning of each one.

For example, the tweet '*Tesla Autopilot Can Navigate On SnowCovered Road With No Lane Markings*' represented by a matrix (number of words x number of layers) equal here to (30 x 30). The graphical representation of the corresponding embedding matrix, using Matplotlib[55], is illustrated in figure 3 by a set of plots where the flat part corresponds to the added "UNK" token and the other part corresponds to the actual words written in the tweets. Other examples of graphical representation of embedding matrices of tweets are given in figure 3

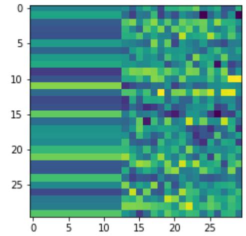
3.3.2 Internal Word Embeddings:

One other approach into Word embeddings is to attach an embedding layer[56] directly into neural network. But before that the data must be organized and put it in the right format for DRNN/CNN architectures, following a two steps procedure:

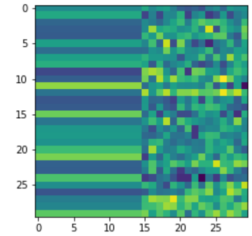
1. Tweets have different lengths, but the maximum length that a tweet can have is 30 words. In consequence, all the tweets in the dataset need to be padding in order to have the same length and to be effectively used. For example, let's consider the tweet '*Tesla Autopilot Can Navigate On SnowCovered Road With No Lane Markings*' this tweet which consist only of 8 words in order to feed it to the DRNN/CNN, is padded with the token(word) 'UNK' and is transformed into '*UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK Tesla Autopilot Can Navigate On SnowCovered Road With No Lane Markings*'.



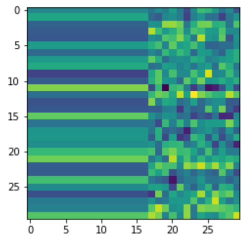
(a) *'Tesla Autopilot Can Navigate On SnowCovered Road With No Lane Markings'*



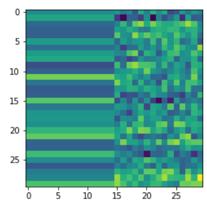
(b) *Tesla wont reveal how it knows its car batteries arent tainted Daily Republic*



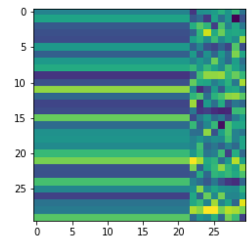
(c) *Norwegian lawsuit says Tesla s horsepower claims are horseshhh well you get the picture*



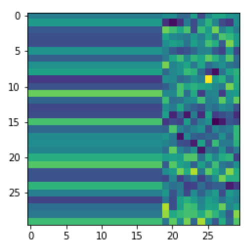
(d) *Volkswagen GTI Turbo Charged One Year Limited Warranty Included Large Awesome Selection Low Prices.*



(e) *Jaguar Land Rover models face NHTSA probe for rollaway risk Some Land Rover SUVs and Jaguar.*



(f) *These Are the Cars Owners Regret Buying the Most*



(g) *Now Honda makes some of the best cars on the road*

Figure 3: Plot representation of the matrix of the original Texts

2. Now that the tweets dataset are transformed into a uniformed dimension and their words need to be tokenized (i.e. attribute each word a number) in order to feed them to the DRNN/CNN. The tokenization method[13] that we used is the classical Bag Of Words[51].

After this process all the tweets of dataset are transformed into a sequence of numbers as showing in the example of Table 3. By applying this procedure to all the selected tweets, these have now the right format to start training the constructed models.

[13] The way to do the tokenization is not important because the embedding layer will be in charge in finding the semantic and contextual inter-tweet relationships. The processes of tokenization need to be similar to an injection [57] i.e. each word need to have a unique encoding and each encoding needs to correspond to one word.

3.4 Comparative Neural Networks Architectures applications for impact sentiment extraction

In order to have the robust way to apply the SSFE method to achieve an efficient predictive result, various neural network architectures were tried. This section gives the 4 of the most performent ones:

- A DRNN with an embedding by Skip-Gram.
- A DRNN with an embedding attaches directly.
- A CNN with an embedding by Skip-Gram.
- A CNN with an embedding attach directly

The metrics used in this comparative are the R^2 score[14] and the $F1$ score[15]

[14] R^2 The coefficient of determination is given by determining the ratio of explained variation over the total variation, it is a statistical measure on how close two datasets are[58].

[15] $F1$ score is a measure of a test's accuracy, it's a weighted average of all the correct positive results divided by the number of all positives(precision) and the correct positives divided be the number of all samples(recall).[59]

Original Tweet	Tesla Autopilot Can Navigate On SnowCovered Road With No Lane Markings
Transformation by padding the Tweet	UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK UNK Tesla Autopilot Can Navigate On SnowCovered Road With No Lane Markings
Tokenization by assigning each word a number	X= [314 314 314 314 314 314 314 314 314 314 314 314 314 314 314 314 314 314 314 350 26 296 236 312 112 49 152 333 52 12]).

Table 3: Example of transforming and tokenization of a selected tweet

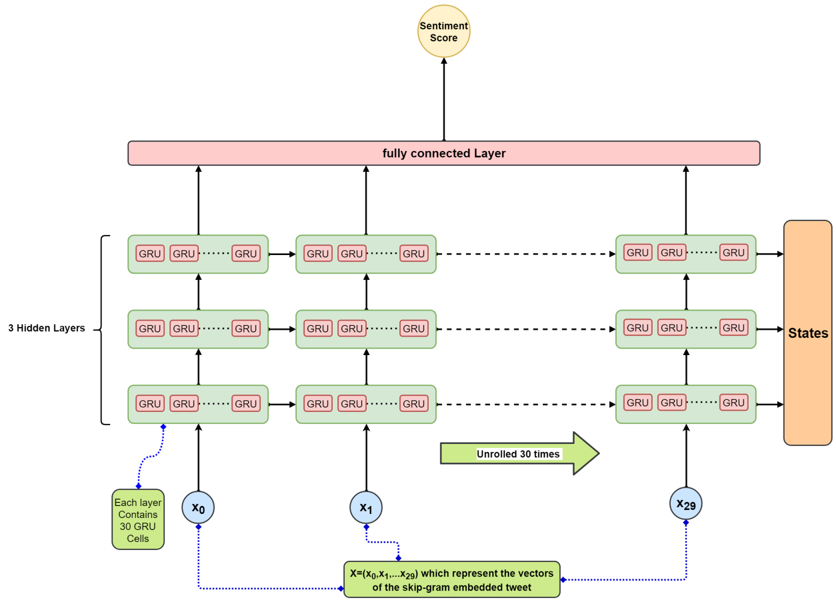


Figure 4: A sketch of the DRNN Model

3.4.1 Deep Recurrent Neural Network with an embedding by Skip-Gram

The majority of DRNNs have a massive amount of neurons and layers. In the case of the study carried out here and as specified in section 3.2, the texts of the tweets treated have a length, in number of words, which does not exceed 30. The resulting embedding matrices for a vector representation of the text of each tweet are easy-to-process dataset. It is then enough of a DRNN, with 3 layers each having 30 layers, to exploit and to lead to very satisfactory results.This architecture, illustrated in figure 4, generates 30x30x30 GRU cells attached to one fully connected layer with a single output. This DRNN architecture gives a success percentage of 99.6% with R^2 score (see 3.4.5 for more details).

The same architecture was also used with 3 outputs instead of one. This allows to calculate the $F1$ score (equal here to 60.5: see section 3.4.5 for more details).

3.4.2 Deep Recurrent Neural Network with an attached embedding layer

Instead of using an embedding dictionary, it is possible to attach an embedding layer directly to the architecture described above (section 3.4.1) and let the model choose the vector embedding as illustrated in Figure 5.

The embeddings are constantly getting update with every back-propagation which makes them very hard to visualize. Using this type of architecture, results in much greater training speed because it is not necessary to construct and to train an embedding model or to create an embedding matrix, then to store its values and reload them again for the final training. In this architecture every calculation is done simultaneously in live memory using CPUDRAM or GPUDRAM.

3.4.3 Convolutional Neural Networkwith an embedding by Skip-Gram

Like previously did for the DRNN, an embedding dictionary is created and tranformed by a study sentences into embedding matrices whose dimensions are equals to : $SentenceLenght \times EmbeddingVectorSize$ (this corresponds in this study to 30x30).

Various activations functions and number of layers were tried and the following process was settled for the architecture described in table 4:

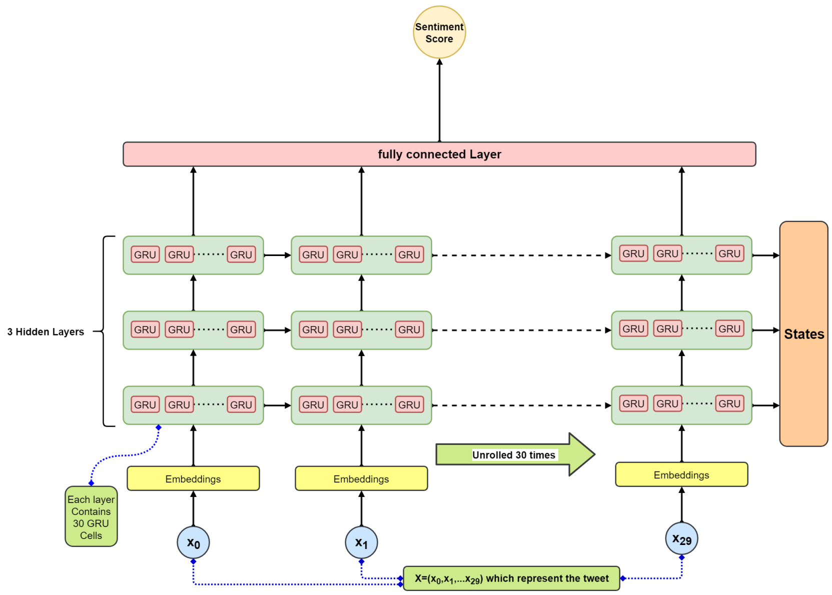


Figure 5: A sketch of the DRNN Model

Layer	Type	Maps	Size	Kernel Size	Stride	Activation
Out	Fully Connected	-	1 $\mid$ 3	-	-	-
hidden1	Fully Connected	-	128	-	-	Relu
Conv2	2D Convolutional Layer	12	30×30	5×5	1	Relu
Conv1	2D Convolutional Layer	6	30×30	5×5	1	Relu
Input	Input	1	30×30	5×5	-	-

Table 4: CNN external embedding Skip-gram Model Architecture

Layer	Type	Maps	Size	Kernel Size	Stride	Activation
Out	Fully Connected	-	1 | 3	-	-	-
hidden1	Fully Connected	-	128	-	-	Relu
Conv2	2D Convolutional Layer	12	30×30	5×5	1	Relu
Conv1	2D Convolutional Layer	6	30×30	5×5	1	Relu
Embed	**Embedding Layer**	**1**	30×30	-	-	-
Input	Input	1	1	-	-	-

Table 5: : CNN with internal embedding architecture

- feed a convolutional layer to another convolutional layer,
- use a fully connected layer with 1 output neuron in the first part,
- apply the same architecture with 3 output neurons.

A fully connected layer with 1 output neuron is used for the calculation of R^2 score and obtains 77.8%. Then the same architecture with 3 output neurons is applied for the calculation of the $F1$ score and gives 64.5%, (See section 3.4.5).

3.4.4 Convolutional Neural Network with attach Embedding layer:

The steps to build this architecture are exactly similar to the steps taken in section 3.4.3. In this case, an embedding layer is directly attached to the CNN model (table 5) and uses the same architecture as above (section 3.4.3).

3.4.5 Result of the Sentiment Analysis:

The following notes are giving here in order to make clearly the result's interpretation:

- Two types of score are used:
  - R^2 score because it best describes the similarities between two vectors.
  - $F1$ score to stay in consistence with the scoring methods used in similar treatments
- The uses data for the SSFE application is cherry-picked i.e. between all the selected tweets, the best representative is easy to distinguish, because it interesting to see: if in the best-case scenario (with a little noise) can a Deep learning model uncover that complicated labeling methods based on psychological principles described in section 3.2, and that it why the precision is a little higher than the state of the art sentiment analysis model [(SemEval2017 [60]),[61]].

A comparison of the 4 architectures seen in 3.4.5, shows that the use of a Recurrent Neural Network with an attached embedding layer is the most stable. While DRNN with detached has higher R^2 score, it has significantly low $F1$ score. Also the Convolutional Neural Network tend to work only in a small interval of Dropouts[62] hyper-parameter " keep probability " i.e. in the first CNN architectures with detach embedding layer, the only interval that gives the best performance is $[0.1, 0.2]$. And in case of the CNN attach embedding layer, the " keep probability " with the most performance is $[0.75, 1]$ which is highly unusual because:

1. The architecture is fairly small.
2. The jump of performances is large.

The Neural Network Architecture	R2 Score(%)	F1 Score(%)
Deep Recurrent Neural Networks	99.6	60.5
Convolutional Neural Networks	77.8	64.5
Deep Recurrent Neural Networks with embedding Layer attach to the network	98.1	64
Convolutional Recurrent neural network with attach embedding layers directly	74.5	62

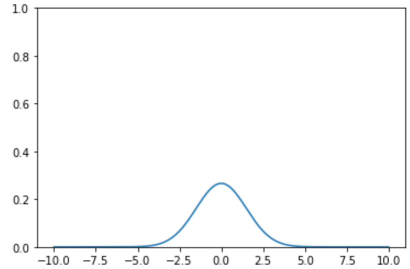
Table 6: he result yielded on the test set after training

3.5 The Forecasts Engine

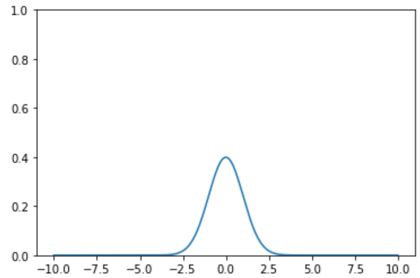
In this step, sentiment analysis are combined with historical sales, using a Sequence to Sequence Recurrent Neural Networks (Seq2Seq RNN).
As reported at the beginning of the article, the application of the SSFE method is focus on sentiment analysis from tweets texts to predict the sales of a car brand model as the Tesla Model S. The originality of this method, based as was shown earlier on implementation of deep-learning, is characterized by consideration of the sentiments of tweets' potential readers. Sentiment analysis is concerned with the impact that a tweet can have and its real influence on the readers.
As stated previously, DRNN with, an attached embedding layer, is the appropriate deep-learning architectures for building and training an in-depth learning model for sentiment analysis of tweets text readers, in the context of car sales forecasts. The results of this model have an accuracy of 98.1% so for each tweet there is a 1.9% chance to miss-classify it, and each tweet has its own volume on interaction (Likes and Retweets).
In this regard it's best to treat the output of the sentiment DRNN predictor with an attached embedding layer as a general stochastic variables rather than absolute truth. To respect this aspect, the SSFE method is based on the application of Equation 1 developed as part of the research work presented in this article.

$$Score_{Model\,Year} = E_{all\,the\,tweets}((1 + Likes_{of\,the\,tweet}) + (1 + Retweets_{of\,the\,tweets}))N(Model_{prediction}, Model_{precision}) \quad (1)$$

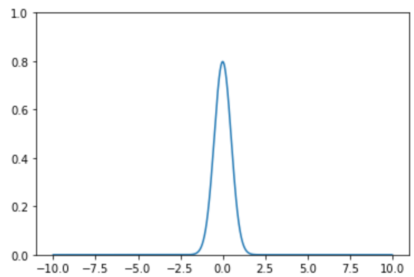
- N is the normal distribution with mean: The Models sentiment prediction and variance: The Models over all test precision.
- $E_{all\,the\,tweets}$ is the mean Operator [63] [64] and it represent the mean over all the tweet gathered regarding that model year.
- $Model_{prediction}$ The prediction of the DRNN in section 3.4.2
- $Model_{precision}$ is the precision of the DRNN in section 3.4.2



(a) Flat distribution



(b) Broad distribution



(c) Compact distribution

Figure 6: Normalized Probabilistic Distribution of the $Score_{Model\,Year}$

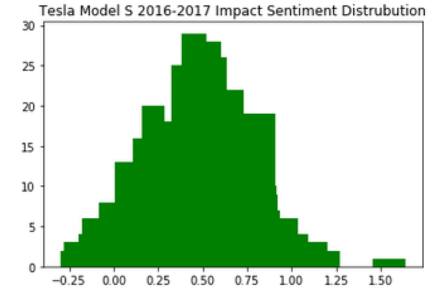
The intuition is: assume that each tweet is a random variable with a Gaussian distribution and consider the average of all tweets. This is the average over all random variables where each random variable is associated with a normal distribution so that the final score calculated in equation 1 is necessarily a random variable and not a scalar. This can lead to the three situations illustrated in figures 6

1. According to the equation 1, if the DRNN model of sentiment analysis is not precise and the interaction of the tweet (i.e. the Likes and Retweets) is high, as a result a non-informative distribution (flat distribution: see figure 6a) which means that SSFE will not try to guess in a non-informative manner.

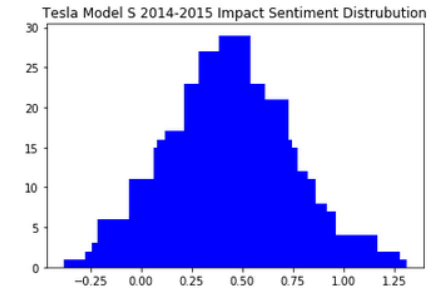
2. If the model is still not precise but the tweet has a minimum interaction, then the distribution will not be flat but rather broad (figure 6b).

3. If the model is still precise but the interaction of the tweet is not noticeable, then the model hesitate to make a prediction because of the lack of the interactions of that group, so the distribution is a broad one (figure 6b).

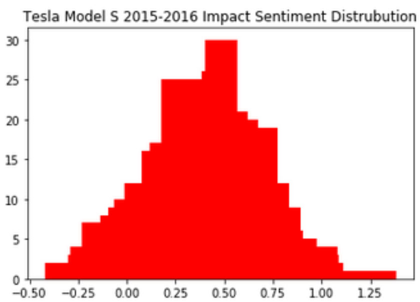
4. On the other side, if the model is precise and can exactly determine the impact-sentiment of a tweet, and that tweet is active, then the distribution will be a narrow distribution (figure 6c). In this case, the model is confident in its prediction.



(a) Sentiment results for the year $2016 - 2017$



(b) Sentiment results for the year $2014 - 2015$



(c) Sentiment results for the year $2015 - 2016$

Figure 7: Shows 100 values extracted from the distrubution $Score_{Model\,Year}$ the X axis represent the numeric values in the vector of step 3 extracted from $Score_{Model\,Year}$ and the Y axis represent their frequency.

As stated at the beginning of the section, the SSFE combine sentiment analysis with a historical sales data. To apply this for the car Market an historical data is provided by kaggle[65]. This, contains more than 90 vehicle models and their sales/registration (registration tends to be 1 to 2 weeks after sales which will not impact the model). The use of these data determines a reference for each vehicle as the one on which the application of the SSFE method is focused. This method can handle any car model, but its application in the framework of this article only concerns Tesla Model S.

- This car model has many competitors, marketed for the same buyer class (the executive class)
- There has been a huge change of perspective in the auto-market where popular brands were forced to change their formulas (e.g big to small engine block, internet connectivity, driver assistance options ...).

The database provided by Kaggle has sales of the Tesla S model over 36 months from 2014 to 2016. This database, as is customary in machine learning, must be divided into two parts:

- 90% of the database i.e. 32 months will be dedicated to training
- 10% of the database i.e. 4 months will be dedicated to testing

The final database that will be used to make the sales predictions is formulated as follow:

1. 2000 tweets retrived on every model year (the application here is on 2014, 2015 and 2016) are retrieved and cleaned then used in the sentiment analysis on the bulk.
2. Equation 1 function is computed to give a score
3. From that distribution, a vector of 100 values is generated. Figure 7 shows the 100 values from the Scores distribution organized by Model Year for the Tesla Model S.
4. Given a month, sales of that month are concatenate with the 100 values extracted on step 3 which is considerate as one vector of 101 variables.
5. The vectors from the previous step are grouped into sequences of length of 8 , which give us 4 sequences for training.

Now the data is formulated to the correct format to start training the Forecast engine.
As stated earlier, a recurrent neural network with basic RNN cells is used With "Relu" (rectified linear unit) activation function by using the following structure of the data :

- Input = [the constructed sequences from step 5]
- Output = [the sales in the constructed sequence in step 5 padded 4 months in the future]

In retrospect, the forecast model is trained to predict 4 months given the last 8 months. The figure 8 exposes the graphic of the result of Smart Sales Forcast Engine.

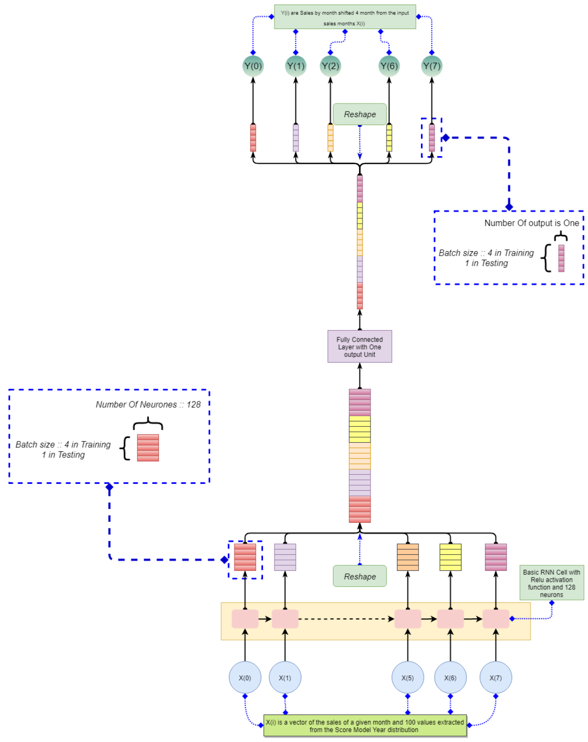


Figure 8: The architecture of the forecast engine

This kind of architecture is also known as sequence to sequence RNN The results yielded by SSFE method are illustratedin the figures 9 and the R^2 score is indicated in Table 7.

As stated earlier, 90% of the available data for 32 months are used for training and 10% corresponding to 4 months are used for testing.
During the training phase, the SSFE managed to successfully predict the values of the real data, but since the used data is a time series the most important test is on the test dataset.
The difference between the real value vector and the SSFE predicted value vector over the 4 training months is just 78 units. However, figure 9 shows that the behavior of the two curves, which corresponds to actual sales and predicted sales, is similar. Thus, the most interesting behavior of predictive outputs by applying the SSFE method is that they have perfectly managed to capture the sales trends (this explains the good R^2 score).

The fact that SSFE predict the increase and decrease in each month, proves that the sentiment impact model managed to extract the overall sentiment about the Tesla Model S and that the sequence to sequence RNN managed to encapsulate those results (the results of the impact sentiment DRNN) with the historical data to make an adequate prediction. This behavioral convergence, between the real and the predicted, shows that a difference with 78 units can only be attributed to external factors. One more important remark is that the difference between actual sales and the predicted values is positive which means that the SSFE method has given a conservative prediction that is reflected in equation 1 which was the desired output.

R^2 score for the Test data with a Sequence to Sequence Reccurent Neural Network	65.4%
R^2 score for the Test data with SSFE	93.3%

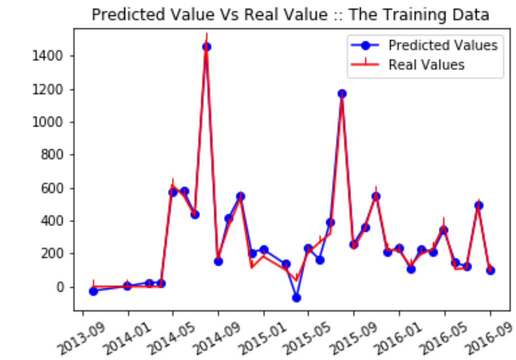
Table 7: The R^2 score on the sales test set

	Number of sold units in the last four months	Number of predicted sold units in the last four months	Difference
Month 1	156	111	45
Month 2	284	264	20
Month 3	43	82	-39
Month 4	170	118	52
Total	**653**	**575**	**78**

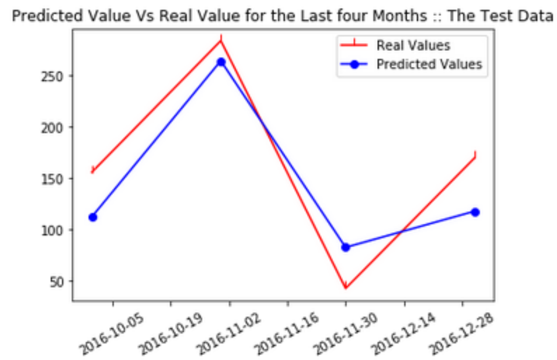
Table 8: The actual number of units predicted by the SSFE and sold

In the recent years the default state of the art of time series predictions became the sequence to sequence recurrent neural network. So, in order to show the utility of the SSFE method, first multiple sequence to sequence RNN are trained using LSTM, GRU, Sigmoid[16] and Relu activation function using only historical sales data, provided by Kaggle. The best architecture gave us an error of 0.357 (following the R^2 scoring) on the test dataset (see table 7) and used it as a benchmark. Then, SSFE is apply to the same data by combining impact-sentiment DRNN, the equation 1 and sequence to sequence RNN. The SSFE method managed to cut the error rate by 80% to only 0.07 (following the R^2 scoring).

[16]Nonlinear activation function, used to be highly popular after the first AI winter now are replaced with Relu, inspired by the firing pattern of cats brains, the mathematical formulation is a s follows $Sigmoid(x) = \frac{1}{1+e^{-x}}$.



(a) Training results



(b) Test results

Figure 9: Graph of the result of the SSFE final output

4 Conclusion

In the above an approach of a new method called "Smart Sales Forecast Engine" (SSFE) for establishing a sales prediction engine was proposed. TheSSFE originality is to take into consideration the analysis of tweets readers' sentiment, which responds to a specific context of a product category. The context on which was built the implementation of the approach illustrated throughout this article, is concerned on the cars sales and especially the purchase predictions brand Tesla Model S. The above study is interested in sentiment analysis based on two psychological principles in relation "Norms of Human Behavior" and "Human Decision" in order to label a collection of carefully selected tweets.
The work thus formed makes it possible to embed the words by applying the most relevant Deep-learning Architectures which allows to build the right model and which gives good results after the training of this one. In the case of the comparative study outlined in this article, this is the DRNN architecture with an attached embedding layer which gave a convincing result considering both applied metric R^2 and $F1$.The result obtained is very positive and is explained by the fact that the application of SSFE, at this stage, has focused on a specific class of potential buyers.
To enable the prediction engine to offer a forecast of car sales, the model developed during the Word Embedding phase in relation to the sentiment analysis was combined with the sales history of the Tesla S model from 2013 to 2016 in Norway, on which was again applied a sequence to sequence RNN.
The sales forecasts obtained agree in their variation with reality but have certain differences. These require improvements that will be placed in the context of prospects for consolidation and evolution of the SSFE method to a global generic engine capable of handling the sales predictions of all types of products. For that, several areas for improvement are clearly identified, namely:

- Reinforce certain aspects of the application of SSFE:
  - Extend the application of the method and its learning by Deep-Learning by taking in consideration users profiles.
  - Use an Auto-Encoder architecture that determines groups, based on social media profiles, and creates a set of data for each of them.
  - Take into consideration the texts written by the automotive professionals' critics.
- Consider some contextual aspects:
  - Take into account the laws and legislation in some areas that could influence sales (environmental protection, CO2pollution, driving ban diesel in some cities, subsidies to encourage rolling clean).
  - Analyze the economy as an attribute that influences the average salary, and therefore the purchasing capacity of the consumers.
  - Take into consideration competitors' offerings.
  - Take an interest in the options of the car models and their amount.
- Extend the study of the device of the SSFE method to other brands and models of cars and more generally to any type of products with a reliable purchase history.
- Take into consideration other types of texts published and exchanged on the Internet through the electronic media, portals and various social networks related to the sector, category or brand of a specific product in order to analyze the feelings of profiles related to their potential readers.

5 References

References

[1] S. E. Shukri, R. I. Yaghi, I. Aljarah, and H. Alsawalqah, "Twitter sentiment analysis: A case study in the automotive industry," in *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Nov 2015, pp. 1–5.

[2] R. M. A. Baracho, G. C. Silva, and L. G. F. Ferreira, "Sentiment analysis in social networks: a study on vehicles," in *Proceedings of Joint V Seminar on Ontology Research in Brazil and VII International Workshop on Metamodels, Ontologies and Semantic Technologies, Recife, Brazil, September 19-21, 2012*, ser. CEUR Workshop Proceedings, A. Malucelli and M. P. Bax, Eds., vol. 938. CEUR-WS.org, 2012, pp. 132–143. [Online]. Available: http://ceur-ws.org/Vol-938/ontobras-most2012_paper11.pdf

[3] Deepthi Duddempudi Mohindra, "Twitter sentiment analysis on car brands," https://towardsdatascience.com/twitter-sentiment-analysis-on-car-brands-c13d449eb9fc, 2017, accessed: 2017-09-30.

[4] J. Garside, "Child labor uncovered in apple's supply chain," The Guardian Fri 25 Jan 2013 19.22 GMT, Accessed 2017-06-27.

[5] "Apple reports fourth quarter results," Apple news room, October 28 2013, accessed 2017-06-27.

[6] "Twitter," https://en.wikipedia.org/wiki/Twitter, Accessed on 2018-02-07.

[7] J. Rooney, "The world's most influential cmos 2017," Forbes June 22 2017, Accessed on 2017-09-05.

[8] "Statistica," https://www.statista.com/statistics/200002/international-car-sales-since-1990, 2018, accessed: 2018-Feb-07.

[9] R. M. Montoya, R. S. Horton, J. L. Vevea, M. Citkowicz, and E. A. Lauber, "A re-examination of the mere exposure effect: The influence of repeated exposure on recognition, familiarity, and liking." vol. 143, pp. 459–498, Accessed on 2017-09-05.

[10] T. Chenier and C. Winkielman, "Mere exposure effect," Accessed on 2017-09-05.

[11] L. K. Fazio, N. M. Brashier, B. K. Payne, and E. J. Marsh, "Knowledge does not protect against illusory truth." vol. 144, pp. 993–1002, Accessed on 2017-09-05.

[12] C. M. . Kelley and D. Lindsay, "Remembering mistaken for knowing: Ease of retrieval as a basis for confidence in answers to general knowledge questions," *Journal of Memory and Language*, vol. 32, no. 1, pp. 1 – 24, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0749596X83710016

[13] R. Pascanu, Ç. Gülçehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *CoRR*, vol. abs/1312.6026, 2013. [Online]. Available: http://arxiv.org/abs/1312.6026

[14] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang, "Recent advances in convolutional neural networks," *CoRR*, vol. abs/1512.07108, 2015. [Online]. Available: http://arxiv.org/abs/1512.07108

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

[16] A. Gatt and E. Krahmer, “Survey of the state of the art in natural language generation: Core tasks, applications and evaluation,” *CoRR*, vol. abs/1703.09902, 2017. [Online]. Available: http://arxiv.org/abs/1703.09902

[17] K. Bandara, C. Bergmeir, and S. Smyl, “Forecasting across time series databases using long short-term memory networks on groups of similar series,” *CoRR*, vol. abs/1710.03222, 2017. [Online]. Available: http://arxiv.org/abs/1710.03222

[18] “Why you feel what you feel,” TEDx Talks TEDxOxford https://www.youtube.com/watch?v=h-rRgpPbR5w, Accessed on 2017-09-11.

[19] T. D. Kemper, “How many emotions are there? wedding the social and the autonomic components,” *American Journal of Sociology*, vol. 93, no. 2, pp. 263–289, 1987. [Online]. Available: http://www.jstor.org/stable/2779585

[20] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez, and W. Nzoukou, “A survey paper on recommender systems,” *CoRR*, vol. abs/1006.5278, 2010. [Online]. Available: http://arxiv.org/abs/1006.5278

[21] S. Zhang, L. Yao, and A. Sun, “Deep learning based recommender system: A survey and new perspectives,” *CoRR*, vol. abs/1707.07435, 2017. [Online]. Available: http://arxiv.org/abs/1707.07435

[22] K. Bontcheva, G. Gorrell, and B. Wessels, “Social media and information overload: Survey results,” *CoRR*, vol. abs/1306.0813, 2013. [Online]. Available: http://arxiv.org/abs/1306.0813

[23] A. Sharma, J. M. Hofman, and D. J. Watts, “Estimating the causal impact of recommendation systems from observational data,” *CoRR*, vol. abs/1510.05569, 2015. [Online]. Available: http://arxiv.org/abs/1510.05569

[24] Z. Wang, Y. Zhang, S. Y. M. Lee, S. Li, and G. Zhou, “A bilingual attention network for code-switched emotion prediction,” in *COLING*, 2016.

[25] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, “Emotional chatting machine: Emotional conversation generation with internal and external memory,” *CoRR*, vol. abs/1704.01074, 2017. [Online]. Available: http://arxiv.org/abs/1704.01074

[26] “Emotion classification,” https://en.wikipedia.org/wiki/Emotion_classification, Accessed on 2017-10-07.

[27] L. Zhang, S. Wang, and B. Liu, “Deep Learning for Sentiment Analysis : A Survey,” *ArXiv e-prints*, Jan. 2018.

[28] C. Olah, “Understanding lstm networks,” colah’s blog, Posted on August 27, 2015, accessed: 2017-08-26.

[29] C. Peng and G. D. Magoulas, “Sequence processing with recurrent neural networks,” in *Encyclopedia of Artificial Intelligence (3 Volumes)*, J. R. Rabuñal, J. Dorado, and A. Pazos, Eds. IGI Global, 2009, pp. 1411–1417. [Online]. Available: http://www.igi-global.com/Bookstore/Chapter.aspx?TitleId=10424

[30] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. O’Reilly Media, Inc., 2017.

[31] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[32] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: http://arxiv.org/abs/1406.1078

[33] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *CoRR*, vol. abs/1503.04069, 2015. [Online]. Available: http://arxiv.org/abs/1503.04069

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *CoRR*, vol. abs/1608.01807, 2016. [Online]. Available: http://arxiv.org/abs/1608.01807

[36] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *CoRR*, vol. abs/1611.10012, 2016. [Online]. Available: http://arxiv.org/abs/1611.10012

[37] Y. Yin, Y. Song, and M. Zhang, "Document-level multi-aspect sentiment classification as machine comprehension," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2044–2054.

[38] X. Zhou, X. Wan, and J. Xiao, "Attention-based lstm network for cross-lingual sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 247–256.

[39] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1014–1023.

[40] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1650–1659.

[41] "A project written in python to get old tweets," https://github.com/Jefferson-Henrique/GetOldTweets-python, Accessed on 2017-09-11.

[42] E. D. Liddy, "Natural language processing," 2001.

[43] "Bag-of-words model," https://en.wikipedia.org/wiki/Bag-of-words_model, Accessed on 2017-06-11.

[44] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 513–520.

[45] S. Zhai and Z. M. Zhang, "Semisupervised autoencoder for sentiment analysis." in *AAAI*, 2016, pp. 1394–1400.

[46] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[47] Z.-Y. Dou, "Capturing user and product information for document level sentiment analysis with deep memory network," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 521–526.

[48] A. Muromägi, K. Sirts, and S. Laur, "Linear ensembles of word embedding models," *CoRR*, vol. abs/1704.01419, 2017. [Online]. Available: http://arxiv.org/abs/1704.01419

[49] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[50] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning.* Association for Computational Linguistics, 2012, pp. 1201–1211.

[51] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[52] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[53] "Vector representations of words,," www.tensorflow.org/tutorials/word2vec, Accessed on 2018-02-07.

[54] "What is one hot encoding? why and when do you have to use it?" https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f., Accessed on 2017-05-07.

[55] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[56] Z. Li, R. Kulhanek, S. Wang, Y. Zhao, and S. Wu, "Slim embedding layers for recurrent neural language models," *CoRR*, vol. abs/1711.09873, 2017. [Online]. Available: http://arxiv.org/abs/1711.09873

[57] R. G. Bartle, "The elements of real analysis," 1964.

[58] "R2 score," http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html, Accessed on 2017-05-15.

[59] "F1 score," https://en.wikipedia.org/wiki/F1_score, Accessed on 2017-05-15.

[60] "Sentiment analysis on tweets task 4," alt.qcri.org/semeval2017/task4/, Accessed on 2017-11-25.

[61] J. Barnes, R. Klinger, and S. S. i. Walde, "Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets," *arXiv preprint arXiv:1709.04219*, 2017.

[62] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[63] "Normal distribution addition," /mathworld.wolfram.com/NormalSumDistribution.html, Accessed on 2017-10-22.

[64] "Affine transformation of gaussian distribution," onlinecourses.science.psu.edu/stat461/node/74, Accessed on 2017-10-22.

[65] "New car sales in norway," https://www.kaggle.com/dmi3kno/new-car-sales-in-norway-eda/data, Accessed on 2017-04-15.