UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) –
BarcelonaTech

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

Automatic generation of comments on twitter based on news

Carlos Casar Morejon

Director: Javier Bejar
Departament: Ciències de la Computació

Grau en enginyeria informàtica

Especialitat: Computació
22 de Gener 2018

Abstract

Being able to draw conclusions from a text and being capable to have an opinion of a text like a human does is still an open problem nowadays.

This project is about the design and implementation of an artificial intelligence that is capable of making new and well-structured comments on Twitter based on the latest news. Using cutting-edge technologies like neural networks and classificatory algorithms we will build a system capable of doing the task. Also we would do a lot of research in finding and creating data sets for the project.

We will show and explain in this document all the technologies, the experiments and the research done in this project.

Resumen

Ser capaz de sacar conclusiones de un texto y poder tener una opinión de un texto tal y como lo haría un humano es hoy en dia un problema abierto.

Este proyecto trata sobre el diseño e implementación de una inteligencia artificial que es capaz de hacer comentarios nuevos y bien estructurados en Twitter utilizando las últimas noticias. Se usaran las últimas tecnologías como redes neuronales y algoritmos clasificatorios para construir un sistema capaz de llevar a cabo la tarea. También se dedicaran esfuerzos en la búsqueda y creación de conjuntos de datos para el proyecto.

En este documento se mostrarán y explicarán todas las tecnologías, experimentos y la investigación hecha en el proyecto.

Resum

Ser capaç de treure conclusions d'un text i poder tenir una opinió d'un text tal com ho faria un humà és avui dia un problema obert.

Aquest projecte tracta sobre el disseny i implementació d'una intel·ligència artificial que és capaç de fer comentaris nous i ben estructurats al Twitter utilitzant les ultimes notícies. Es faran servir les últimes tecnologies com xarxes neuronals i algoritmes classificatoris per construir un sistema capaç de dur a terme la tasca. També es dediquessin esforços en la recerca i creació de conjunts de dades per al projecte.

En aquest documents es mostraran i explicaran totes les tecnologies, experiments i la investigació feta en el projecte.

Contents

1 Context and scope of the project

1.1 Context and problem formulation

In our modern society, the information is playing an important role. We receive tons of information every day by many different devices. In addition to the classic media like television and radio, we have now social media [1] [2]. Nowadays an important part of the citizens get the news from social media and in this fast-paced society, it is important to have good resumes of the news.

As some studies show [3], the time spent reading news online is pretty low in average compared to print news. To keep the attention of the users, the news must be short enough to be fully read by the users, that is the reason that we need good resumes.

One of the most trendy fields in the past years in computation is Machine Learning, a branch of Artificial intelligence. The reason for his popularity is the power and scalability on it. On the last few years, Machine learning has been used in a wide range of applications[4][5][6].

Machine Learning allows the machine to learn and make decisions by itself. It takes large amounts of data to do that, that is why Machine Learning is the answer to the computation problems where the explicit programming is not possible. That is the reason why in this days, the information age, Machine Learning is a trend.

The objective of this project is to create an AI with machine learning techniques, to be able to comment news on twitter. To perform that, first, the application will use the classification method to categorize the news. Secondly, and most important, we will create a neural network connecting news and comments. To finish it we will create an algorithm that each 10 minutes takes the last news from a specific newspaper and writes his opinion about it.

1.2 Actors

The development of the project involves several actors, which are listed and described in the following lines.

Developer: Is the person in charge of research, implement and document the whole project. In this project, there is only one developer, myself, and I am the person in charge to accomplish the deadlines.
Director: Is the main responsible for guiding and giving advice to the developer. In this project, Javier Bejar, from the Computer Science department, has acted as director.

Beneficiaries: The main beneficiary of this project is an online newspaper, that wants a bot that comments news by itself. Also, it can be the interest of different kind of researchers working in related topics or using similar techniques.

1.3 State of the art

To put it briefly, machine learning is a subject undergoing intense study, so there is a lot of research and improvements going on these days. Since machine learning has become a thing, almost every field has been affected, from healthcare to financial trading to smart cars. Drawing conclusions from texts has not been an exception [7], being capable of analyze large amounts of data has always being important.

One example of a real application on web forum retrieval and text analysis [8]. With this, we can replicate and train our machine with real data on how humans behave on the internet. Doing a good text analysis is a very important thing, because all the information that the program will need comes from there.

Maybe the most similar approach to what is being proposed in this project are the chatbots, that are capable of drawing conclusions from a text [9], that after each sentence draws a conclusion and answers. This kind of chatbots functions very similar to our project, the only difference is that we focus on news, but the process of drawing a conclusion of a text is very similar. But, what are chatbots? Chatbots are computer programs that mimic conversation with people using artificial intelligence. These chatbots are widely known, and almost every company has one, like Alexa from amazon to google Allo.

Also, there are the bots for Twitter, with a wide range of them. For example there is **@metaphorminute** that is a Twitter bot that every minute creates a new and original metaphor. Another bot of Twitter is **@netflix_bot**, that tweets new releases on Netflix instant. One of the most famous is **@DearAssistant**, that will try to answer your questions just like Siri, Google Now or Cortana. As we can see there is a lot of diversity of Twitter bots regarding his functionality.

1.4 Objectives

The main objective of our project is to develop a tool that is capable of creating new and original comments of news on Twitter. We aim to design the program in a way that the sentences of the comments make syntaxes sense.

Another objective for this project is to have a high accuracy with the machine learning algorithms, in order to avoid critical mistakes like classify a sports news into a political news. A high accuracy will reduce the possible errors and will make easy figure out where the mistakes are made.

An additional objective is to find several data sets for the training of the classification algorithm. This is an important part of the algorithm because a good data set will carry on to good results.

1.5 Scope

In order to be able to solve the problem that defines our project first, we will program a machine learning code to clasificate news, from sports to politics to science. First, we need to find a good amount of data, to train our machine. That data has to contain, at least, the text of the new and the data type. Also the data has to come from an official and reliable source, like a university[9]. If we don't find an appropriate set of data, we will have to train with the deep learning technique, so the machine will classify the data only with the text.

One more important algorithm is the communication with Twitter, that will enable us to collect data and Tweet our comments. For this algorithm we will use the Tweepy library for accessing the Twitter api. As we will explain in the comment algorithm, this library will help to the creation of comments, collecting tweets that comment news.

When our machine is able to recognize different types of news, and we are able to communicate properly with twitter, we will proceed with the comments. First, we need to create another machine learning code, this time training with tweeters that usually comment news on twitter, using the Tweepy library that we mention before. These tweets will be classified in the same way as the news are, therefore, we will have for each type of new, many different answer. With all of this data collected, we will try to generate a comment for a new, first classifying it, and then, generating the comment.

1.6 Methodology

Since the timetable is really tight, the project will be developed in an agile methodology. This fast and flexible methodology will help us develop the project.

By using an agile method, we will be able to review regularly to have the project in good control. Furthermore, we will fix little goals each week to accomplish, in order to keep the project fresh and not digress from the goal

Now we will show an outline of how we worked on the project:

- First we learned machine learning, and thinking the best implementations for our algorithms (Classification, Regression, Clustering...). Also we learn what are we capable of doing with this technologie and doing some research on similar problems and his results.
- Once we learned enough machine learning, we start to search for a good dataset for our first algorithm and then we implement it.
- Also, we have to implement the communication with twitter, to at least be able to Tweet every x minutes.
- After that we implemented the comment generator, with a previous search and some learning in deep learning. For this algorithm, we will have to find another dataset but this time with comments on news.

- Once the algorithm works, we will do some test and continue to improve the algorithm until it fulfils our goals.

1.6.1 Development tools

For the development of the main program we will use Python, with his library tweepy, that allows us to work with the api of Twitter. Additionally, we will use many math and machine learning libraries, to make the coding and algorithms easier. The main reasons for using Python above other high level languages is his deep documentation about twitter bots and machine learning.

The monitoring tool to keep track the development of the project is going to be github. This will improve the communication about the code between director and developer.

1.6.2 Validation of results

To check if the program is working as we want, we will have periodical meetings with the director to check the results. Also we will do many test on the results until we think the comments are close to what an human can write.

1.7 Possible obstacles of the project

1.7.1 Data sets

One of the main problem we can have with the project is not finding the appropriate data sets for training our machine. If this occurs, we would probably change to a deep learning approach, where the datasets we need are easier to find.

1.7.2 Classificatory Algorithm

The main obstacle that we can find while working on this algorithm is reaching a high accuracy classification the topics of the news.

1.7.3 Comment Algorithm

The comment generator algorithm is probably the hardest task of this project. It will need a lot of accuracy in order to make sentences that make sense, and that means a lot of work making an optimized code.
Another possible problem for the algorithm is to find a good data set with appropriate comments and news. Additionally, these comments have to be of high quality, which makes this task even more difficult.
Also. we could also find some problems creating comments with only machine learning, more than likely we should help the construction of sentences with other rule based algorithms.

1.7.4 Bugs

Considering that the machine learning framework is complex, and we will have long algorithms, we will make unit test to ensure that there is no bugs.

2 Temporal planning

This section will describe the tasks that are going to take place in the project and the time to accomplish them. An action plan will be provided in order to finish the project in the desired time. However, the initial planning could be revised and modified as a result of the evolution of the project.

The estimated project duration is of about 4 months and a half, starting in 1st September, 2017 and the deadline is on 21st January, 2017.

2.1 Task description

2.1.1 Learning machine learning

The first step toward this project is getting a wide background in machine learning, due to is the core of the algorithms in this project. I started by reading this online book[10], that gave me a general view and the basics to understand the topic. Afterwards, I continue with an online practice tutorial[11], that introduced me on the programming part of the machine learning.
This process of learning took a month for the reason that learning from 0 machine learning is not an easy task.
This task did not required any material resources, but it did required human resources to read and understand all the information about machine learning.

2.1.2 Project planning

This is the task covered by the GEP course, that defines what is going to be done and how is going to be done in this project. It can be divided in the following three stages:

- Context and scope of the project
- Project planning
- Budget and sustainability

This task did not required any material resource(except the material about the GEP course in atenea), but it did required human resources to research, write and understand all the information about machine learning.

2.1.3 Tweepy library

Tweepy is a Python library for accessing the Twitter API. This library is core for our project, because we need to retrieve information for the learning process of our machine and also Tweet our comments. Our main source for learning how to use this library is the official documentation for Tweepy[12].

As material resource, the Python idle is needed in order to practice.This task also requires human resources to read and understand all the information about the library.

2.1.4 Data sets

As we already explain in the process to create a machine learning algorithm, we need large data sets, with the proper structure, to train our program to be able to recognize the topic of a new. Searching this data sets is not a simple task, we need some official and reliable repositories from a good source.
This task did not required any material resources, but it did required human resources to read and research all the information about data sets.

2.1.5 Recognize the topics of news algorithm

This algorithm will function as a complement for the comment generator, being a auxiliar source for creating different data set for the neural network. We will see further information of these in section 6.5.
For this first algorithm we will use the classification method, to separate the news by topics. There is many and different algorithms for the classification method, but for this specific project we think that k-nearest neighbors[13] algorithm fits better with our project. With the proper data set, this algorithm will work perfectly, because every new of the same topic has similarities, then this will ensure a high accuracy.
However, if the accuracy of the algorithm is not as high as we want, we will have to try other algorithms, looking for a higher accuracy.
For the datasets, we will try more than one, looking for the best performance, maybe training with more than one at the time.
Once the algorithm is working correctly, we will do some test to prove is working as we want. We will try with some news from newspapers to find if the algorithm decides his topic correctly.
As material resource, the Python idle is needed in order to code.This task also requires human resources to code and understand.

2.1.6 Communication with twitter

As we said in the 2.1.2, we will use the Tweepy library to use the Twitter API. First, we will program a function to Tweet things, for the comment section of the project.
Afterwards, we will program the information retrieval which will be used to relate news and tweets. With this information, we will be able to construct the second algorithm, the comment generator.
As material resource, the Python idle is needed in order to code.This task also requires human resources to code and understand.

2.1.6 Source of information for the Comment generator

This task was not part of the original planification (see 2.5 for further information).

In this task we will search for a big dataset of information in order to train our algorithm, because retrieving information from twitter to train was too difficult and was not enough information to train the bot.

2.1.7 Comment generator

This is the main task of this project, it will be the visible part of the project that shows that all the previous algorithms works. It will be in charge of comment news from a newspaper and Tweet it.
This algorithm has to be very precise, because as we said, it will be the one who generates the comments and any grammatical error or nonsense sentence will be terrible. For this algorithm, we will use a neural network algorithm, that we will explain later in section 5.

As material resource, the Python idle is needed in order to code.This task also requires human resources to code and understand.

2.1.8 Twitter bot

The Twitter bot will be the algorithm that uses the comment generator, first searching for news in some important newspapers and then using the model of the comment generator to create and opinion and tweet it.

2.1.9 Final task

In this task we are going to look over the whole project works as expected and we are going to prepare the delivery of the project, including the documentation and preparing the final presentation.

2.2 Time table

The table 1 is an estimation of the time spent in each task described in the previous section.

Task	Estimated duration(Hours)
Learning machine learning	80
Project planning	80
Tweepy library	20
Data sets	25
Recognize the topics of news algorithm	75
Communication with twitter	25
Source of information for the Comment generator	40
Comment generator	115
Final task	40
Total	500

Table 1: Time table of each task of the project

2.3 Gantt chart

Figure 1 shows a Gantt chart of the different task of the project.

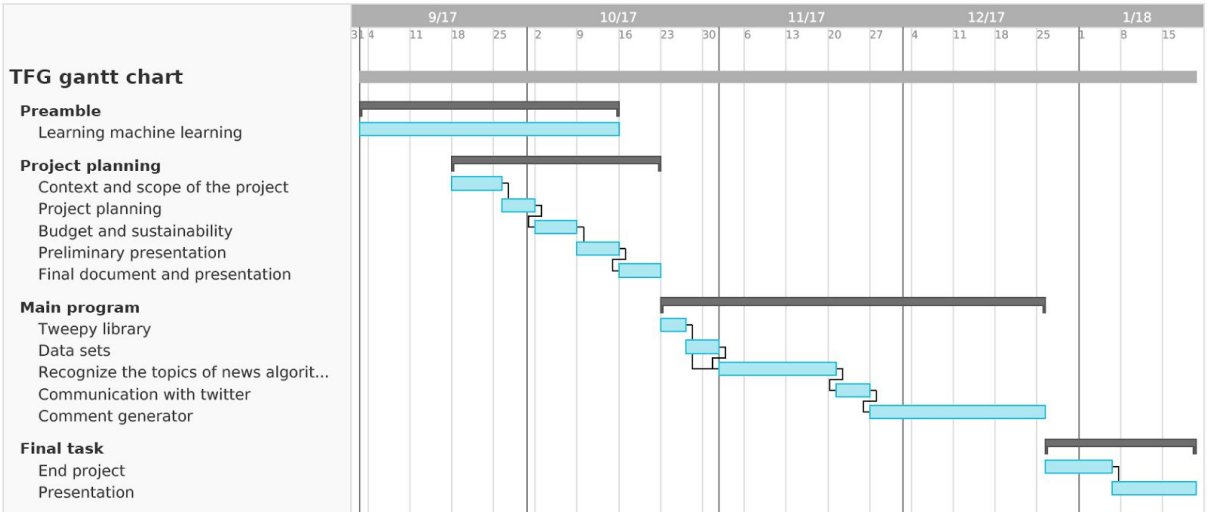

Figure 1: Gantt chart of the project

2.4 Alternatives and Action plan

In this section we are going to describe how are we going to execute the planning that we made in this section.

The idea is to work as we have planned, doing each task at the time we point out, but we know that every project has delays and obstacles that may make difficult to stick to the plan. If the problems occur and we run out of time we are going to try to get only a basic version only. Certainly, as every project we have priorities. We must have a working algorithm in most cases, that can comment any new properly. We will try to have the best accuracy possible in the machine learning algorithms, but having a working program is the goal.

We are going to have meetings with the director every time that an important stage of the project is finished.

Next, some examples of potential sources of delays and alternatives are mentioned.

2.4.1 Sources of information on Twitter

One of the biggest problems that we can have in this project is to find reliable comments on Twitter to build our machine learning algorithm for the comments. This could cause delays until we find a proper source of information or an alternative approach.

2.4.2 Optimization problems

A bad selection of the optimizations to be done could result in a waste of time without a significant gain. This could cause delays, because optimizing the code is fundamental in the development of machine learning codes.

2.5 Deviation from the original plan

Because of a bad initial planning and several complications developing the project, we did have to put more hours, especially on the comment generator. Originally, we planned to retrieve information from Twitter to train the bot but turned out to be too difficult and to have so little information that was impossible to work with that.

Also, we did have put some extra hours to learn some deep learning and how chatbot works in order to make a good comment generator.

3 Budget and sustainability

3.1 Budget estimation

In this section, we are going to do a budget estimation for our project. In this document, we will take into account three resources: Hardware, software and human resources. At the end of the section we are going to show the total budget of the three resources.

To calculate the amortization of each resource, we are taking into account two factors, the first one being the useful life and the second one being the duration of our project, that will go on for 4 months. Also, the usual life duration will be, according to the tax office, 3-4 years for the hardware and 2-3 for the software.

3.1.1 Hardware resources

Table 2 contains the cost and the amortization of the hardware that we are going to use in the project.

Product	Price (€)	Useful life	Amortisation (€)
PC (included all the needed devices)	800.00	4 years	43.83

Table 2: Amortisation and prince for the hardware products.

3.1.2 Software resources

Table 3 contains the cost and the amortization of the software that we are going to use in the project.

Product	Price (€)	Useful life	Amortisation (€)
Windows 10 Pro	260.00	3 years	18.99
Python IDLE	0.00	3 years	0.00
Github	0.00	3 years	0.00
Google Docs	0.00	3 years	0.00
TeamGantt	0.00	3 years	0.00
Total	260.00	-	18.99

Table 3: Price and amortization for the software products

3.1.3 Human resources

This project is going to be developed by one person. For that reason, this person will be the Project manager, Software developer and tester. The 460 hours of the project will be distributed between the 3 roles. In table 4 and estimation of human resources is showed.

Role	Price per hour (€/h)	Time (h)	Cost (€)
Project manager	30	90	2700
Software developer	20	310	6200
Tester	15	60	900
Total	-	460	9800

Table 4: Cost estimation by role.

Following, table 5 provides the exact time that each role spends in the different task of the project that we previously have defined.

Task	Duration (hours)	Dedication (hours)		
		Project manager	Software developer	Tester
Learning machine learning	80	0	80	0
Project planning	80	80	0	0
Tweepy library	20	0	20	0
Data sets	25	0	25	0
Recognize the topics of news algorithm	75	0	55	20
Communication with twitter	25	0	20	5
Comment generator	115	0	85	30
Final task	40	10	30	5
Total	460	90	310	60

Table 5: Time estimation by role and task.

3.1.4 Total budget

Table 6 shows the total cost of the project, using the data shown in tables 4 3 and 2.

Concept	Cost
Hardware resources	800 €
Software resources	260 €
Human resources	9800 €
Total	10860 €

Table 6: Total budget cost

3.2 Budget control

As previously mentioned, our budget will need modifications if we can't follow the established plan.

We could have difficulties in our project, but is improbable that we need more hardware resources aside for the ones already mentioned. We might need more software resources for the development of the project, but there is plenty of free options for the software development.

The most difficult tasks in this project are the main machine learning algorithms, which could take longer to develop. This task involves the software developer and the tester, so we have to take into account that the money spent on them could grow if the problem becomes harder than we expected.

3.3 Sustainability

In this section we are going to measure the sustainability of our project in this three dimensions: Economic dimension, social dimension and environmental dimension. This measure will be based on the application of the sustainability matrix, as shown on table 7.

	PPP	**Useful life**	**Risks**
Environmental	Design consumption	Ecological footprint	Environmental risks
	9/10	16/20	-3/-20
Economical	Bill	Viability plan	Economical risks
	7/10	15/20	-1/-20
Social	Personal impact	Social impact	Social risks
	9/10	14/20	-5/-20
Sustainability range	25/30	45/60	-9/-60
	61/90		

Table 7: Sustainability matrix

3.3.1 Environmental dimension

- **PPP:** The development of this project uses the minimum amount of resources possible, only the electricity required for the PC to work. Therefore, searching alternatives to reduce the consumption is pretty much impossible. Also, the reuse of resources in this project is difficult too. The project will make automatic comments on news by Twitter, which has a very little consumption of electricity.
- **Useful life:** The ecological footprint of the project is pretty low, because is a software project and it only involves the cpu consumption and the consumption of the execution of the program. Also, it is a very difficult task measuring the consumption of the execution of the program.
- **Risk:** The actual environmental risk of the project is that a lot of people uses this program and therefore the consumption of the cpu increases.

3.3.2 Economic dimension

- **PPP:** A detailed budget has been done for this project, including material and human resources as shown in previous sections of this document.

- **Useful life:** The proposed solution will be less expensive than current solutions from an economic point of view, because of the efficiency of the algorithms and his performance it will be cheaper in time and energy for the final user.
- **Risk:** There is not an actual economical risk besides a posible incrementation in the price of the gpu necessary for the development of the algorithm with the neural machine translation.

3.3.3 Social dimension

- **PPP:** The execution of this project has teached me how hard is to develop a big project, and to put some perspective to the problems. Obviously, the project entails the knowledge and improvements in programming machine learning techniques. Also writing this thesis has improved a lot my skills in writing and also in the research of good papers and articles.
- **Useful life:** As mentioned in the introduction section, machine learning is a subject undergoing intense study. For that reason, all the researchers that work on this field could benefit from this project which guarantees an important social impact at least in the investigation field.
- **Risk:** The biggest risk for this project is the job losses of some scholarship journalist that comment news if the comment generator is good enough.

4 Machine learning algorithms

In this section we are going to describe and explain the machine learning algorithms used in the 2 big main algorithms of this project: The comment generator and The news classifier. First we are going to introduce the Deep learning, Natural-language processing(NLP), recurrent neural networks(RNN) Neural machine translation(NMT) that is the algorithm that we used to make the comment generator and finally some knowledge on what is Tensorflow. Secondly, we will explain the algorithm that we chose for the news commentator and why.

4.1 Deep learning and neural networks

Deep learning is about constructing machine learning models that learn a hierarchical representation of the data.
Neural networks are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve performance on) tasks by considering examples, generally without task-specific programming. For example, in our project, they might learn to identify comments that correspond to that news analyzing example of news/comments have been processed before and using the results to create comments for other news. They do this without any a priori knowledge about news and comments. Instead, they evolve their own set of relevant characteristics from the learning material that they process.
Neural networks are part of the deep learning because you can describe a hierarchical model where each layer of neurons represents a level in that hierarchy, and with the power of the GPU's, today we can add tons of layers, that is why is also called Deep Neural Networks(DNN).

4.1.1 Basic Neural Network

We can see Neural Networks as an directed acyclic graph where each node is a function and each connection . The basic form of a Neural Network has a set of nodes X called input layer and another set of nodes B called output layer. For each node A_i there is $|B|$ connections to each B_i where this connections have a weight W_{ij}. The value of each node B_i comes represented by $g((\sum_{j=1}^{|B|} W_{ij}A_j) + b_i)$ where b_i is a real number independent of any input named *bias* and g a function $\Re \Rightarrow \Re$.

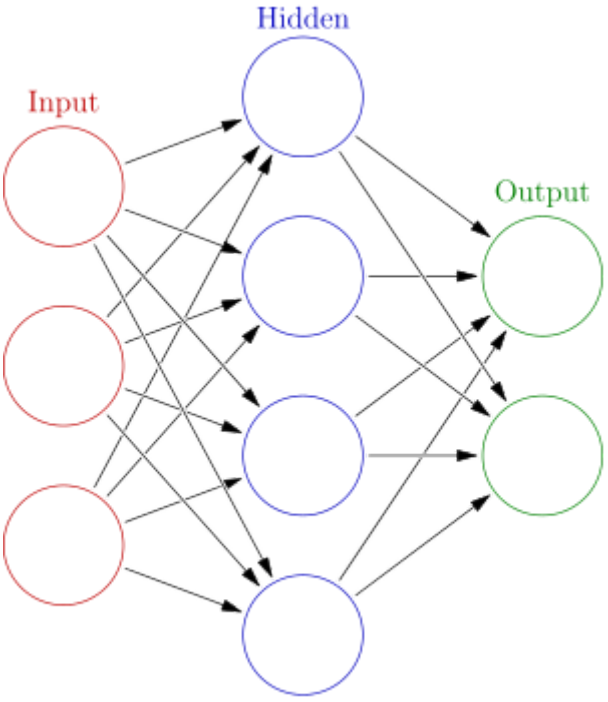

Figure 2: A simple example of a basic Neural network with three layers (Inputs, Hidden and output)

As we said before, nowadays with the technologie that we have we can add tons of layers, having a lot of different *neurons* which leads to a very powerful and expressive method.

4.1.2 Training of a basic Neural Network

There is a lot of variants for the training of neural networks, but the most important one consist in a minimization of a function named gradient descent.

This technique consist in a iterative method based on the gradient value of the function. Given a model M with a vector of parameters V, an initial approximation, a set for training T of pairs (x_i, y_i) and a function F(V) that evaluates the error of the model over the set of training T, updates every parameter V as $V_j = V_j - \alpha \frac{\partial F}{\partial V_j}(V)$ (where α is a real number) until reaching a fixed point regarding the error function or until fulfils a determined convergence criteria. It is important to point out that this method does not guarantee a global minimum.

In the case of neural networks there is two points to take into account: The initial weights of V and the calculation of the gradient. Regarding the initial weights, the most common practice is to assign aleatorie values within an interval, and is highly suggested not to assign uniform values, because it can cause local minimums in some functions. The partial derivatives are calculated obtaining the result of activating the network with an initial value (x, y) and then for each node, calculating the error and the corresponding partial derivatives with the corresponding weights.

Neural networks, like other methods of machine learning can suffer of overfit and underfit. Underfit is given when the function calculated for the neural network does not work well with examples out of the training set. This is a common thing when the architecture is to big and complex regarding the length of the data set. On the other hand we have the underfit, that as opposed to overfit, it happens when the architecture is not enough expressed regarding the function to be calculated.

4.1.3 Recurrent Neural Networks

The idea behind RNNs is to make use of sequential information. In a traditional neural network we assume that all inputs (and outputs) are independent of each other. But for many tasks that's a very bad idea. If you want to predict the next word in a sentence you better know which words came before it. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations.

Given a sequence $x_1, ..., x_n$ and one node a the output for x_i is calculated as $y_i = g(Wx_i + Uy_{i-1} + b)$ where W and U are matrix of weights and y_0 is an state already defined (usually it is initialized as a vector with all the values different from 0)

4.1.4 Training on Recurrent Neural Networks

The training on the recurrent neural networks will use the same algorithm explained in section 5.1.2, appling it repeatedly in every element of the sequence.

4.2 Natural language processing (NLP)

Natural-language processing is a field of artificial intelligence concerned with the interactions between computers and human (natural) languages, and, in particular, concerned with programming computers to fruitfully process large natural language data. Following we are going to introduce some of the most important tasks in natural language processing that are used in this project:

- **Parsing:** Determine the parse tree (grammatical analysis) of a given sentence. The grammar for natural languages is ambiguous and typical sentences have multiple possible analyses. In fact, perhaps surprisingly, for a typical sentence there may be thousands of potential parses (most of which will seem completely nonsensical to a human). There are two primary types of parsing, Dependency Parsing and Constituency Parsing. Dependency Parsing focuses on the relationships between words in a sentence (marking things like Primary Objects and predicates), whereas Constituency Parsing focuses on building out the Parse Tree using a Probabilistic Context-Free Grammar (PCFG).
- **Word segmentation:** Separate a chunk of continuous text into separate words. For a language like English, this is fairly trivial, since words are usually separated by spaces.

4.3 Introduction to sequence-to-sequence (seq2seq)

Sequence-to-sequence learning (Seq2Seq) is about training models to convert sequences from one domain (e.g. sentences in English) to sequences in another domain (e.g. the same sentences translated to French).
This can be used for machine translation or for free-from question answering (generating a natural language answer given a natural language question) -- in general, it is applicable any time you need to generate text. In this project, we are going to create a natural language, where the natural language question are the main corpus of the news and the natural language answer are the best comments/opinions for these news.

4.3.1 General case translating sequence-to-sequence

In the general case, like in our project ,input sequences and output sequences have different lengths and the entire input sequence is required in order to start predicting the target. Following we are going to explain how the training works:

- A RNN layer acts as "encoder": it processes the input sequence and returns its own internal state. Note that we discard the outputs of the encoder RNN, only recovering the state. This state will serve as the "context", or "conditioning", of the decoder in the next step.
- Another RNN layer acts as "decoder": it is trained to predict the next characters of the target sequence, given previous characters of the target sequence. Specifically, it is trained to turn the target sequences into the same sequences but offset by one timestep in the future, a training process called "teacher forcing" in this context. Importantly, the encoder uses as initial state the state vectors from the encoder, which is how the decoder obtains information about what it is supposed to generate. Effectively, the decoder learns to generate targets[t+1...] given targets[...t], conditioned on the input sequence.

In inference mode (when we want to decode unknown input sequences), we go through a slightly different process:

1. Encode the input sequence into state vectors.
2. Start with a target sequence of size 1 (just the start-of-sequence character).
3. Feed the state vectors and 1-char target sequence to the decoder to produce predictions for the next character.
4. Sample the next character using these predictions (we simply use argmax).
5. Append the sampled character to the target sequence
6. Repeat until we generate the end-of-sequence character or we hit the character limit.

The same process can also be used to train a Seq2Seq network without "teacher forcing", by reinjecting the decoder's predictions into the decoder.

4.3 Introduction to Neural Machine Translation (NMT)

Sequence-to-sequence (seq2seq) models have enjoyed great success in a variety of tasks such as machine translation, speech recognition, and text summarization. We will introduce seq2seq models applied to a Neural Machine Translation.
Formerly, traditional computational translation systems performed their task by breaking the source text into chunks of works and then translated them word-by-word. This has been proven to be a bad translation, and not like we humans do translate. We read the entire source text, understand its meaning and then we reply with a conclusion. NMT mimics that process to do better translations.

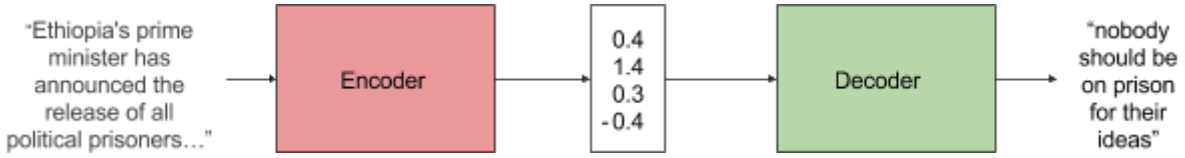

Figure 2. Encoder-decoder architecture – example of a general approach for NMT. The encoder converts the news into a meaning vector which is passed over a decoder to obtain a translation

Specifically, an NMT system first reads the source text using an encoder to build a "thought" vector[20], a sequence of numbers that represents the sentence meaning; a decoder, then, processes the sentence vector to emit a translation, as illustrated in Figure 2. This is often referred to as the encoder-decoder architecture. In our case, it will translate News into comments, an English to English translation.

NMT models vary in terms of his architecture, but for sequential data they mainly use a recurrent neural network (RNN). Usually an RNN is used for both the encoder and decoder. Also, there are several models of RNN determined by: Directionality, Depth and Type. Now we are going to explain some of them and which one fits better to the project.

4.4 Introduction to Tensorflow

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

NMT models execute on Tensorflow, as a result we will have access to an important tool called TensorBoard.

4.4.1 TensorBoard

Once we have our model trained, we want to know how good it is, and besides checking manually asking the model to inference an answer, we have tensorboard. Tensorboard it is a suite of visualization tools made to help us debug, understand and optimize our tensorflow models. Tensorboard has many metrics, that we are going to analyze which ones are good for our project and how to analyze them.

- **Bleu metric:** Bleu is probably the best determining factor on how good a translation was, however in our project we are not really translating, when we do comments over news, we don't have only one or two good answers like when we translate from spanish to english, we can have infinite good comments over news. Despite that is not the most important one, we would like to see it increase a little bit while we train.
- **Learning rate:** For this variable we want to see it decreasing overtime.
- **Train loss:** Train loss will decrease and with some punctual increases overtime. When it starts to climb again we should stop training the model.
- **Perplexity:** This variable is a probability distribution, how far off you are. Opposite of Bleu metric, we want to see perplexity to decrease, maybe into single digits.

Also, we can use TensorBoard to compare training runs, collect runtime stats, and generate histograms.

4.5 Simple RNN

In a simple RNN, the model receives the words embeddings(I) of the original text that we want to translate. This sequence of words embeddings is now processed by two recurrent neural networks: The encoder(E) RNN simply consumes the input source words without making any prediction; the decoder(D), on the other hand, processes the target sentence while predicting the next words. Once this process finish, we decode the answer obtained into words, obtaining the final answer(A).

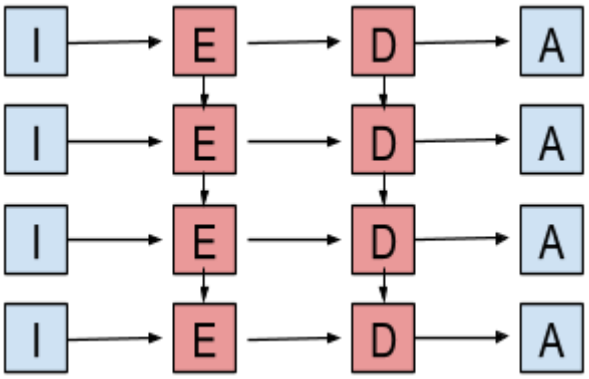

Figure 3: Architecture of a simple RNN model

The main idea behind this architecture is to have units that know what it has been read until now (unit E) and other units that knows what it has been translated (unit D). This architecture fits very well with normal translation between languages, like French to English, were the

structure of the language could be different, and it has better results that only translating word by word without taking into account the other previous words in the sentence. Nonetheless, we are not translating a language into another language, we are translating news into comments, that means a big chunk of information to another, normally less longer, and only knowing what it has been translated before is not enough for this task, we need to get the full meaning of the sentence in order to do a good comment.

4.6 Bidirectional RNN

The bidirectional RNN is an extension of what we have seen in section 6.2. This time, the RNN of the encoder goes both ways creating two vectors, one reads the sequence up to down and the other one down to up, then we link these 2 vectors creating again only 1 vector E.

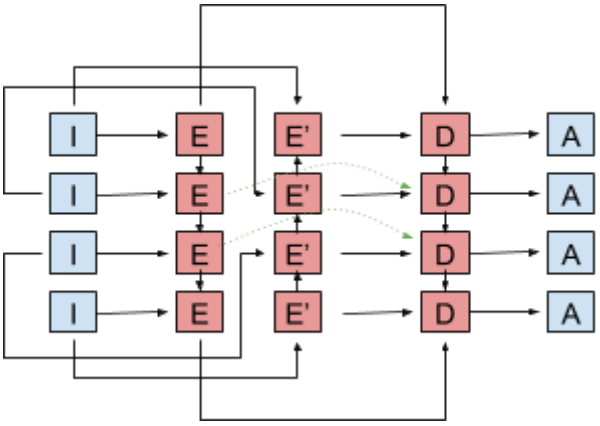

Figure 4: Architecture of the bidirectional RNN model

With this in mind, we figure out that this model fits better our project, because of the reason that reading in both ways is reading all before drawing a conclusion, knowing all the information is very beneficial.

4.7 Attention Mechanism

The attention mechanism is an optimization of what we have seen in the section 6.3 with the Bidirectional RNN. The key idea of the attention mechanism is to establish direct short-cut connections between the target and the source by paying "attention" to relevant source content as we translate. The attention mechanism does that by creating a kind of dynamic memory containing every hidden step before the last source state from the encoder. By doing that, the translation of longer sentences improves a lot, because you have every middle step, that you can focus on what is important, not only the last source state from the encoder. Also, this mechanism takes into account the last decoded state.
In figure 5 we can see an example of an attention mechanism, where AM is the attention mechanism that stores all the encoder mid steps and also the last decoded state.

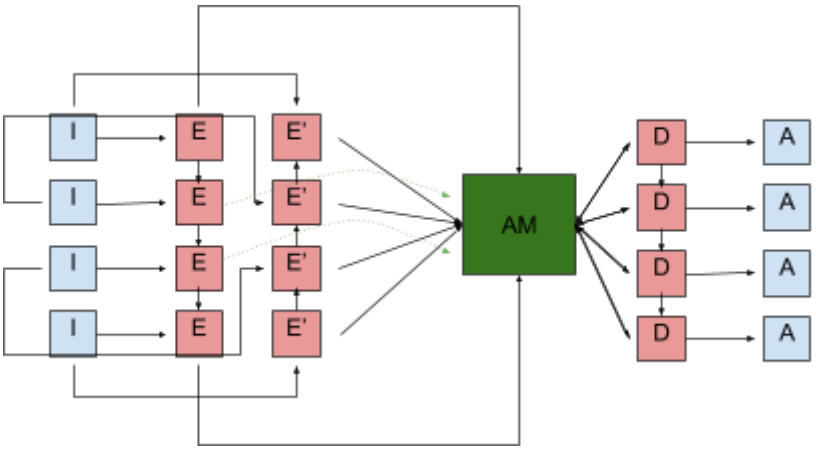

Figure 5: Example of how the attention mechanism works.

4.8 Classification algorithms

Classification algorithms have enjoyed great success in classifying many different types of data, but there is not only one algorithm for classifying data, and each one is better at some task than the others.
We have chosen two different algorithms to solve the problem: Naive Bayes and Support Vector Machine(SVM). We did chose these two algorithms because their are the most used when trying to build a classifier.

4.8.1 Support Vector Machine (SVM)

The SVM is one of the most popular and used machine learning algorithms to classify. This algorithm is a binary clasificator, that means that in a vector space, with n points (where n is the number of elements you want to classify), were the value of a element belongs to a specific coordinate. The SVM will find the best separating hyperplane that will separate the data in 2 groups like we see in the figure 6.

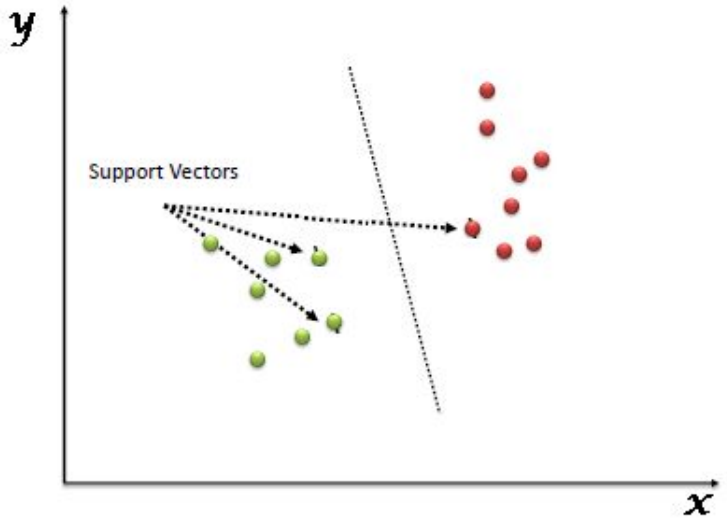

Figure 6: Example of a hyperplane separating the data in a vector space.

The best separating hyperplane is the one that has the longest margin with the nearest point in the plane. We can see in Figure 7 three different hyperplanes A, B and C, and the best one is the hyperplane C, that has the longest margin with the closest point of the plane.

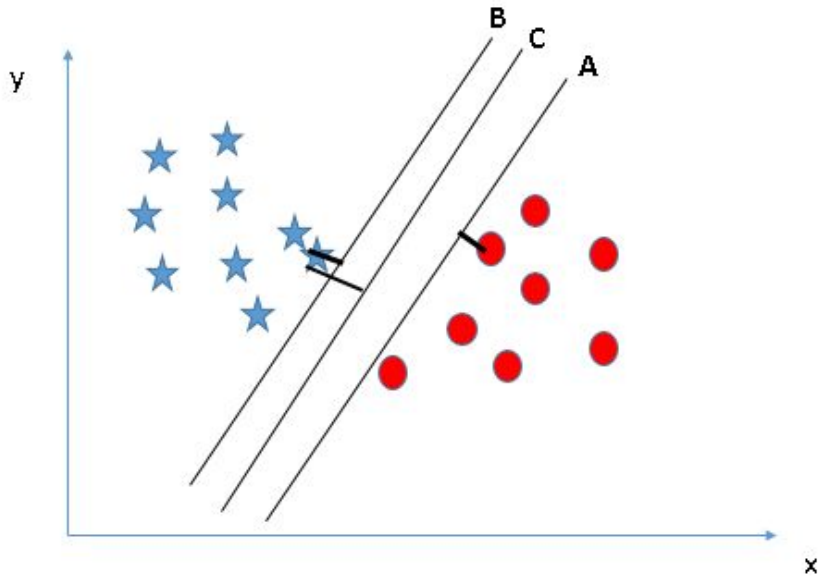

Figure 7 Example of three hyperplane showing the closest margin of each one

Also we did make a list of the positives and the negatives about using SVM:
Pros:

- It works really well with clear margin of separation
- It is effective in high dimensional spaces.
- It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Cons:

- It doesn't perform well, when we have large data set because the required training time is higher
- It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping

4.8.2 Naive Bayes classifier

Naive bayes it is not a single algorithm but a family of algorithms that all share a common principle, that every feature being classified is independent of the value of any other feature. So for example, a fruit may be considered to be an apple if it is red, round, and about 3″ in diameter. A Naive Bayes classifier considers each of these "features" (red, round, 3" in diameter) to contribute independently to the probability that the fruit is an apple, regardless of any correlations between features. Features, however, aren't always independent which is often seen as a shortcoming of the Naive Bayes algorithm and this is why it's labeled "naive".

We can now replace the fruits by news, where the features are its words and the number of times they appear.

Also we did make a list of the positives and the negatives about using Naive Bayes:

Advantages:

- It’s relatively simple to understand and build
- It’s easily trained, even with a small dataset
- It’s fast!
- It’s not sensitive to irrelevant features

Disadvantages:

- It assumes every feature is independent, which isn’t always the case

5 Datasets and database description

In this section we are going to describe all the data that we used in this project. All the data that we used in this project comes from two public data set and one data set that we have retrieved. The first one is a reddit data set posted by Stuck_In_the_Matrix [15]. This data set contains every public comment on reddit and also every public source of reddit since 2006, but we are only using a little portion of them. To be more precise, we are only using the months of June, July, August, September and October from 2017. The purpose of this data set is training a NMT(Neural Machine Translation) with tensorflow to be able to comment news. Also, there is another data set from Stuck_In_the_Matrix that contains every post on reddit. We used this data set to retrieve every new from June to October of 2017 and relation the content of the news with the comments we talked before.
The second one is a bbc data set for news[16], that we are going to use to create a machine learning classification that will help in the process of training of the commentary algorithm.

5.1 Subreddits

As we now, reddit has many different subreddits each one for a different topic to discuss. Because our project tries to comment on news, we will focus only in the subreddits that comment news, selecting only a few of them: News, WorldNews, Entertainment, Economics, science, PoliticalDiscussion and moderatepolitics.

- **News subreddit:** This is the main subreddit for news of all kind and from every country, will be our main source of information for the training of the program.
- **WorldNews subreddit:** Another big subreddit for news, but this time does not include the USA news.
- **Entertainment subreddit:** The Entertainment subreddit focuses on news of the entertainment industry(mainly USA, but also other countries).
- **Economics subreddit:** The most important subreddit to discourse about all kind of research, news, academic work on news.
- **Science subreddit:** Science is the main subreddit to talk and discourse about all the breaking news on many different areas of science (medicine, neuroscience, social science, biology, …).
- **PoliticalDiscussion subreddit:** This is one of the most serious subreddits about politics, where no jokes are able and every comment has to be substantial and has to contribute to the discussion.
- **moderatepolitics subreddit:** Very similar to politicalDiscussion subreddit but not so sober.

5.2 BigQuery and data composition

Storing and querying massive datasets can be time consuming and expensive without the right hardware and infrastructure. Google BigQuery is an enterprise data warehouse that solves this problem by enabling super-fast SQL queries using the processing power of Google's infrastructure.

We used BigQuery to only take the info that we need, that is much more smaller, considering we only need some specific subreddits and some specific fields from the JSON. For that reason, we did twitter different queries to obtain two different tables, one for the source news of reddit[17] and the other one for the comments[18]. Next we will describe the restrictions we put into the BigQuery in order to avoid post or comments we did not want in our data set and we are going to describe the two tables that we finally obtained after the restrictions:

5.2.1 Source_Reddit

It contains all the info from the source subreddits that we need:

Fields that we selected from the query:

- **id:** It is the unique identificator of each post of reddit. This id is composed of 6 letters/numbers like: “745qf1”.
- **subreddit:** This field shows the subreddit of the post.
- **created_utc:** Indicates the time when the post was created. We will use this in case we need some kind of order.
- **url:** This field is the most important one because it has the url of the new, that we will extract later.

Restrictions of the query:

- **subreddits**: We filter to obtain only the following subreddits: (News, WorldNews, Entertainment, Economics, science, PoliticalDiscussion and moderatepolitics)
- **Number of comments**: We did also filter to only retrieve the posts that have one or more comments, avoiding posts with no comments, because it will be useless for the reason that we will not have any comment to connect with the news.

5.2.2 Comments_Reddit

It contains all the info from the comments of the subreddits that we need:

Fields that we selected from the query:

- **id:** It is the unique identificator of each comment on reddit. This id is composed of 6 letters/numbers like: “745qf2”.
- **parent_id:** This id references the id of the parent reply for this comment, in this case, it will be always the id of the post. This id is composed of 6 letters/numbers like: “745qf2”.
- **subreddit:**This field shows the subreddit of the post.
- **body:** It contains the text of the comment.
- **score:** It is the score of the comment, the higher, the better.
- **created_utc:**Indicates the time when the post was created. We will use this in case we need some kind of order.

Restrictions of the query:

- **subreddits**: We filter to obtain only the following subreddits: (News, WorldNews, Entertainment, Economics, science, PoliticalDiscussion and moderatepolitics)

- **Erased comments:** Filtering for erased comments is important, because it would ruin the data set with comments that are not real.
- **Only comment reply to parent**: For this restriction we wanted to guarantee that we only have comments that reply the parent post, not comments replying other comments of the same post. This will prevent having comments that did not talk about the news but about the comment that other users made of this news.

Once we did all this pre-work on the datasets, we did go from 350 Gigabytes to 200 Megabytes, we saved a lot of space and also a lot of useless data to process, increasing the speed and the performance of the algorithm that handles the data.

5.3 Processing data

In this section we are going to deal with the two JSON we talked in the previous section. We are going to create a database that contains the information of both JSON, but instead of the url, we are going to retrieve the information of the corresponding new and put it on the database.

5.3.1 Database description

The database will contain the following fields:
parent_id(Primary key): It will be the unique id of the news. This id is composed of 6 letters/numbers like: “745qf1”.
comment_id: It will be the unique id of each comment. This id is composed of 6 letters/numbers like: “745qf1”.
parent: This field will contain all the information of the news.
comment: It is the comment of the pertinent news
subreddit: It is the subreddit of the post and of the comment.
unix: The field unix is the time on which the comment was made.
score: It is the score of the comment, the higher, the better.

5.3.2 Transforming url to data

The source data of reddit only gave us the url of the new, nevertheless we need the body of the news. In order to do that text extraction we used a python library called Newspaper [19] for each url of the JSON file.

This process took a long time, one of the main reasons was that it was a lot of data to process and scraping directly from the web one at the time tooks a lot of time. Also, there were some url that were not working, mainly because of the website not working or because the Newspaper library did not recognize the webpage and it could not retrieve any information from it.

5.4 Transforming the database onto training data

Finally, once we have all our database complete, with all the news paired with his best comment, we have to transform this database in to training data. In order to do this we followed the recommendations that google gives in his tutorial for seq2seq [21].

1. **Generate data in parallel text format:** In order to generate our data in parallel text format we have decided to create two files. The first one is train.from, where each line of this file will contain the corpus text of the news. The second file is train.to, where each line of this file will contain the comment of the news of the corresponding line number of the file train.from. To sum up, each line of the file train.from will have his corresponding answer in the file train.to in the same line number.
2. **Tokenize your data:** In order to tokenize the data, we will tokenize by words,
3. **Creating a vocabulary:** We will create two vocabularies, one for each train file, based on the most common words. The length of the vocabulary of 15.000 words for each vocabulary, but it may change during the experiments as we will see in section 6.
4. **Learn and apply subwords units to handle rare and unknown words:** We will use this for example in url of web pages, or if we want to common expressions to be only one word.

In addition to train.from and train.to, we did two much more smallers files, containing the 5% of the total rows. These two new files are test.from and test.to, will be used in the training section of the NMT to give us feedback on how is the training going.
In total, we have about 100.000 paired rows, with the news and his best comment. This is not a big data set, but is enough to start working on it and obtaining decent results.

5.5 BBC dataset

For the news classifier algorithm we were looking for a dataset containing the most important types of news. First, we found the 20 newsgroups text dataset, that is one of the most famous datasets in order to train a classifier for news. This dataset has 20 different topics, that is a point against it, because as we said before we were looking one only with the most general ones.

Also, we find after training a little bit, that the dataset was outdated, and the actual news are quite different from that ones, and it was not behaving correctly.

After doing more research we found the BBC dataset. This dataset is divided in 5 categories(business, entertainment, politics, sport, tech) that are the most general type of news in the newspapers and also, the news in the dataset were much more actual than the news on the 20 newsgroup dataset.

Each file only contains the news, we don't need further information because each news is in his corresponding folder, and when we read them we already know which topic is each news based on the folder containing them.

Also, a little portion of this news are divided in a training section, taking around the 5% of the news. We do this for later on, on the experiments section, we can see how good is our algorithm doing, like the percentage of accuracy. With that information we can do small changes to the algorithm and see if the changes improve the accuracy of the algorithm or not.

6 Experiments

In this section we are going to describe the different experiments and stages of the development of the project. First, as in section 6, we are going to start with the experiments on the comment generator algorithm, then we'll talk about the experiments done in the news classifier.

For all the experiments of this section we are going to use the files train.from, train.to, test.from and train.to that we already explained in detail in section 5.4.

Additionally, in all the experiments of this section will be used the attention mechanism explained in section 5.4, because it has been proven much more effective than the others [22].

6.1 Pre setup of the NMT

Before we start the training of the comment generator, we have many thinks we have to set up like vocabulary blacklist, answers blacklist, answers detokenize etc. This files will be constructed with regular expressions.

- **Vocabulary blacklist:** This file will contain every word that we don't want to be in our vocabulary, like insult or common expression on the forum reddits that we don't want.
- **Vocabulary replace:** Like in the vocabulary blacklist one, this file will affect the vocabulary but instead of erasing words here we can replace the words we don't want for other words that we think that are better.
- **Answers blacklist:** In this file we will have every answer that we don't want to reproduce.
- **Answers detokenize:** Here we want to detokenize the sentence, like web url or removing unnecessary spaces.
- **Answers replace:** This time replace the answer with synonyms instead of blacklist them.
- **Protected phrases:** Ensures that matching phrases will remain untouched when building vocab file

All this files will improve the training and also, as we train, if we see some recurrent sentences we can replace or blacklist them.

Additionally, we have some parameters that are important to check. The values of this parameters will change the performance of the training, and its important to find the best values with prove and error sometimes.

- **Vocabulary size:** Defining the vocabulary size is important, it will determine how many words are available for the nmt to use as answers. His size will be determined by the memory we have and how many words we think it fits better. A huge

vocabulary size can be bad too, as we put many words that are often unused or simply wrong writed. We started with a vocabulary size of 15.000 words.

- **num_train_steps:** This variable determines the number of steps that the training will make. If its value is ‘none’, the machine will train forever.
- **batch_size:** The batch size determines the number of sentences that the training process executes at the same time. Its value is important when the input sources are very long because that can cause that our machine runs out of memory pretty fast. In our case his value has to be very low, like 5.
- **Optimizator:** The optimizator is the algorithm that trains the model. For this project we are going to use ADAM, which it is the most used optimizator nowadays. ADAM computes adaptive learning rates for each parameter.

6.2 First experiment

For this first experiment we set the values of the parameters as we can see in table 8.

Name of the parameter	**Value**
Vocabulary size	15.000
number training steps	500.000
batch size	2
learning rate	0.001
Optimizer	ADAM

Table 8: parameters for the first experiment

In this first attempt to make it work, we did not get any good results, the training was stopping after only 1000 steps because reaching the memory limit. We did some changes on the values of some parameters like batch size from 128 to 2, and vocabulary size from 20.000. With these changes we reach 5000 steps but was not enough for the next points:

- The perplexity was of 300, to high considering we want it on single digits
- The bleu score was of 0, horrible even if it doesn't have to reach the values of a normal translator from english to french, 0 is not even close of any good results
- Probably the most important one were that the answers were not making any sense, repeating himself or not getting a proper structure.

Even reducing the more important conflicting values with memory(batch size and vocabulary size) we could not get more than 5.000 steps.

6.2 Improving the steps

Seeing that we could not get more steps changin every single parameter of the nmt, we think about changing the datasets. We knew that some news were very long, like more than 1.000 words, and not really adding such important information to the news. With that in mind we decided to put a boundary around 400 words, thats more or less the 2 first paragraphs of the news, that contains all the important information.
Once we did that restriction, we then tried to train the nmt again with the parameters we can see on table 9

Name of the parameter	**Value**
Vocabulary size	15.000
number training steps	500.000
batch size	3
learning rate	0.001
Optimizer	ADAM

Table 9: parameters for the experiment

With this training we started growing the steps beyond 5.000 but another problem occurred: We did have a lot of answers saying the same and were not connected with the news, then we look over the data sets and we saw a patron of answer that was very common done by the moderators.

6.3 Reform of the data sets

Observing that the answers by the mods were very similar and did not tell anything about the news, we tried to find this answers in the data sets and replacing them with proper answers. This process took a long time and we have to do it several times until we replaced all the data that was affecting the training.

Once all the data set was reformed, we restarted the training with the same parameters as we did in section 6.2 on table 9. After some training, we can start looking into the data that tensorboard brings us. This time we can see how the perplexity goes down over time in figure 8.

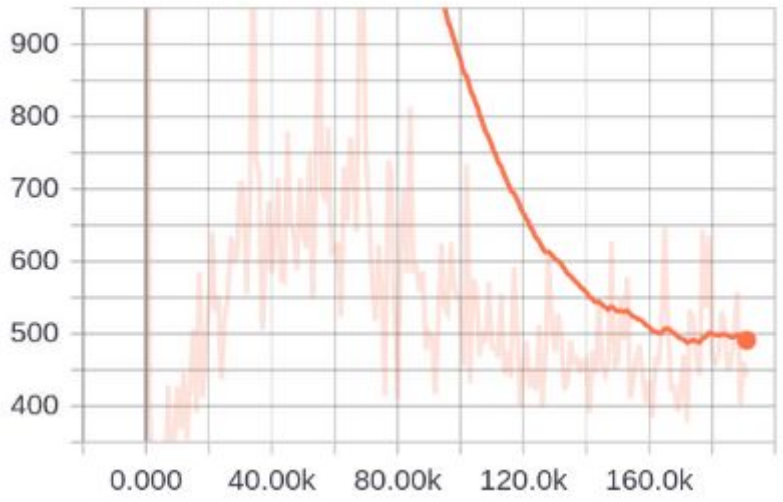

Figure 8: Perplexity dropping over the steps.

Also and more important, we can see how blue is slowly increasing his value as we show in figure 9:

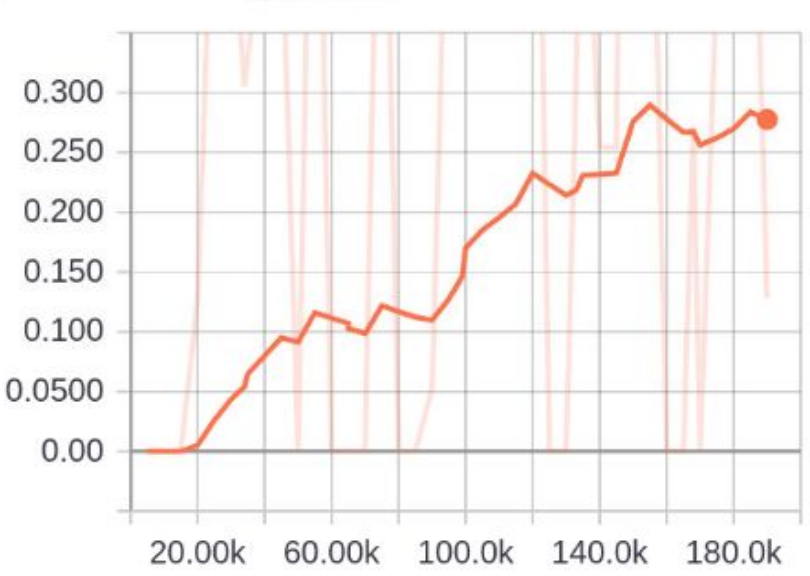

Figure 9: Slowly increasing value of Bleu score over the steps

But not everything looks that good, as we can see in figure 10, the learning rate is not moving at all, therefore that means that our model is too slow learning.

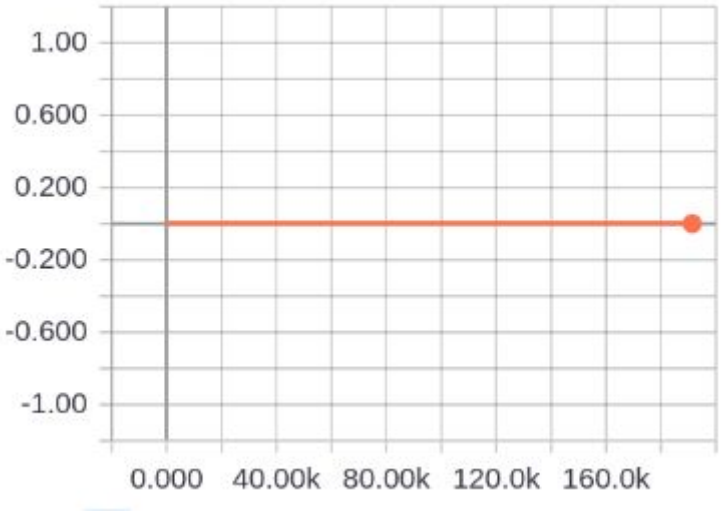

Figure 10: Learning rate over steps

Taking into account this statistics we can say this model that we did learns too slow, and we have thinked two main reasons that could be harming the learning rate of the model:

- The short size of the data set
- The incredibly length of the news data, that also damages the batch_size we can use, because it could run out of memory.

Finally, we can give some examples of the good and bad answers of the system, where **source** is the news text, **reference** is what the real comment was and **nmt** is what our machine answers.

Here we have an posible good answer at the step 65000:
source: *The Trump administration is moving toward handing back to Russia two diplomatic compounds , near New York City and on Maryland ' s Eastern Shore , that its officials were ejected from in late December as punishment for Moscow ' s interference in the 2 0 1 6 presidential election .*
President Barack Obama said Dec. 2 9 that the compounds were being " used by Russian personnel for intelligence - related purposes " and gave Russia 2 4 hours to vacate them . Separately , Obama expelled from the United States what he said were 3 5 Russian " intelligence operatives . "
[The luxurious , 4 5 - acre compound in Maryland being shut down for alleged Russian espionage]
In early May , the Trump administration told the Russians that it would consider turning the properties back over to them if Moscow would lift its freeze , imposed in 2 0 1 4 in retaliation for U.S. sanctions related to Ukraine , on construction of a new U.S. consulate on a certain parcel of land in St. Petersburg .
Two days later , the U.S. position changed . Secretary of State Rex Tillerson told Russian Foreign Minister Sergei Lavrov and Russian Ambassador Sergey Kislyak at a meeting in Washington that the United States had dropped any linkage between the compounds and the consulate , according to several people with knowledge of the exchanges .

1 of 2 1 Full Screen Autoplay Close Skip Ad × Russian compounds in Maryland and New York shut down View Photos Two luxury retreats , in Centreville , Md . , and Oyster Bay , N.Y. , where Russian diplomats have gone for decades to play tennis , sail and swim , were shut down by the Obama administration in retaliation for Moscow ' s alleged hacking in the presidential election . Caption Two luxury retreats , in Centreville , Md . , and Oyster Bay , N.Y. , where Russian diplomats have gone for decades to play tennis , sail and swim , were shut down by the Obama
reference: *Well , they did help him get elected . Fair is fair right ?*
nmt: *I ' m not sure why they ' re not going to do anything .*

But not every answer is good, we get some times answers that repeats himself with the same words or answers that uses <unk>. The <unk> token means that is a word that is not in our vocabulary and by default they do not show the word. One example of both problems is this one:

source: *PORTLAND — Unease about white supremacist activity in Portland deepened after the fatal stabbings of two men who tried to shield young women from an anti - Muslim tirade , and some people worry that the famously tolerant community could see a resurgence of the hostilities that once earned it the nickname " Skinhead City . "*
The attack aboard a light - rail train happened Friday , the first day of Ramadan , the holiest time of the year for Muslims . Authorities say Jeremy Joseph Christian started verbally abusing two young women , including one wearing a hijab . When three men on the train intervened , police say , Christian attacked them , killing two and wounding one .
Court documents released Tuesday for the first time mentioned a fourth man who was the first to intervene and was not attacked , but they did not identify him by his full name .
Christian , 3 5 , was defiant during his brief initial court appearance Tuesday , shouting : " You call it terrorism , I call it patriotism ! " He made repeated outbursts , saying , " You ' ve got no safe place ! " and " Death to the enemies of America ! "
Christian , who faces aggravated murder and other charges , didn ' t enter a plea . He has been appointed public defenders . Lane Borg , head of the local public defender agency , said the office was " saddened by this tragedy " but urged people to let the justice system take its course .
In the probable cause affidavit , prosecutors said video feeds in the back of a patrol car captured Christian saying after his arrest that he had stabbed three people in the neck . His court - appointed attorney , Gregory Scholl , did not immediately return a call for comment .
Long and violent history
The deaths stunned the city , but also underscored nervousness about recent events , including

reference: *Oregonian here ! Back in the 1 9 0 0 ' s , some of the Oregon Legislature were also members of the KKK .*
> Earlier this year , organizers of a small community parade affiliated with the city ' s famous Rose Festival canceled the celebration over fears of violence after protesters said the local Republican Party had plans to allow a " neo - Nazi hate group " to march with them . Local GOP leaders denied the charges .

> In the suburb of Troutdale , an Iranian refugee found his home painted with racist graffiti and death threats . And in Gresham , another eastern suburb , prosecutors charged a man with a hate crime after police said he chased down a black teenager with his car after a fight and struck him , killing him .
> For years , Portland was the home base for Volksfront , a now - defunct white separatist organization founded in 1 9 9 4 , according to the Southern Poverty Law Center , which tracks hate groups .

nmt: I ' m not a fan of <unk> , but I ' m not a fan of <unk> , but I ' m not a fan of <unk> , but I don ' t see it . newlinechar newlinechar I ' m not a fan of <unk> , but I ' m sure it ' s

Here we can see how I'm not a fan of <unk> is repeated constantly.

6.4 Performance of classification algorithms

Here we are going to explain the performances that we obtained using the test files we mentioned in section 5.5 with the Naive Bayes and the Support Vector Machine algorithms. We runned both algorithms over the test files trying to inference the topic of the news and we obtained the following results:

- 82% of accuracy with the Naive Bayes algorithm
- 92% of accuracy with the Support Vector Machine

Taking into account this results, we have decided to continue with the SVM for the next experiments.

6.5 Aborted experiments

In this section we are going to talk about the experiments that we did not do or we aborted and the reasons why we decided not to finish them.

Five different models depending on the news topic: The idea behind this experiment was that the topics of the news got influence on the comments of that news. This experiment could in theory improve the quality of the bleu score and in general of the comment generator. This experiment was planned in early stages of the project and we could not accomplish it for several reasons. This was an ambitious experiment, because of the fact that we need to train five different models, which is an incredible amount of time and resources. Also, doing five separated models we should also do five different data sets and probably retrieving more information than the retrieved for one model which is a really expensive task.

Improving the inference model: The inference model that we use to get the answers is the default one that the NMT has, but we could modify to improve the performance. One way to improve this inference model was to create our personal score. The algorithm will take the 10 best answers that the default inference model gives us. Then, we will create some score rules, to harm the punctuation of the answers that we don't want, like an answer that repeats

himself, by decreasing his score, and favour the sentences we think they are right by upgrading his score. We could not fully implement this due a lack of time to experiment with it to show good results, because of the reason that this solutions requires a lot of prove and error to find the best model.

7 Twitter bot implementation

Finally, we took the model implemented in the section 6.3 and use it on our Twitter bot. This bot is very simple, it will do the following steps:

1. It will take every 10 minutes the latest news from the BBC online newspaper with the newspaper library we already mention in section 5.3.2
2. It will check if we already did a comment on it. If the news is not commentated yet, we use the our NMT model to infer an opinion on the text.
3. Wether step 2 is completed or not, it returns to step 1 to check for more news.

The name of this bot is @c_casar on Twitter.

8 Conclusion

In this project has been developed a neural translation system that is able to comment news as said in section 1.

The first thought is that system developed in this project it is still in development, the experimentation with long text, like summarization with neural networks it still has problems making long sentences. The project has the same problem with long sentences, where his performance drops considerably.

Another point is the collection of information of the project, was a really difficult task, that usually takes long time to have reliable source of data to work with. Also the data set we obtained probably was not long enough for the commentator, but we don't know as there is not other information about this.

Also, the length of the news was to much for the hardware of these days, we did have to drop the length of every news and also the batch size for the training, that did hurt the performance of the algorithm. Maybe in the future, when machines are more powerful this kind of algorithms could work perfectly.

On the other hand we did obtain some decent results, and it is clear that neural networks is a very powerful tool that we will see more in the future.

9 References

[1] “News Use Across Social Media Platforms 2016”
http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/

[2] “Social Media Use for News and Individuals' Social Capital”
http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2012.01574.x/full

[3]”National press online readers average 30 seconds per day versus 40 minutes for print”
http://www.pressgazette.co.uk/study-national-press-online-readers-average-30-seconds-per-day-versus-40-minutes-for-print/

[4] "Machine learning in genetics and genomics - NCBI - NIH."
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5204302/.

[5] "Machine learning applications in cancer ... - ScienceDirect.com."
http://www.sciencedirect.com/science/article/pii/S2001037014000464.

[6] "End to End Learning for Self-Driving Cars." 25 abr.. 2016,
https://arxiv.org/abs/1604.07316.

[7] "Text categorization with Support Vector Machines... - ResearchGate."
https://link.springer.com/chapter/10.1007%2FBFb0026683?LI=true

[8] “Web Forum Retrieval and Text Analytics: A Survey”
http://www.nowpublishers.com/article/Details/INR-062

[9] "A Deep Reinforcement Learning Chatbot"
https://arxiv.org/abs/1709.02349v2

[10] “UC Irvine Machine Learning Repository”
https://archive.ics.uci.edu/ml/datasets.html

[11] "Introduction to Machine Learning - Alex Smola."
http://alex.smola.org/drafts/thebook.pdf

[12] "Machine Learning with Python tutorial series - Python Programming.net."
https://pythonprogramming.net/machine-learning-tutorial-python-introduction/.

[13] "Tweepy Documentation"
http://docs.tweepy.org/.

[14]"A Branch and Bound Algorithm for Computing k-Nearest Neighbors."
http://ieeexplore.ieee.org/abstract/document/1672890/

[15] "I have every publicly available Reddit comment for research"

https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/?st=j9udbxta&sh=69e4fee7

[16] “BBC dataset for machine learning training reasons”
http://mlg.ucd.ie/datasets/bbc.html

[17] “BigQuery reddit posts”
https://bigquery.cloud.google.com/dataset/fh-bigquery:reddit_posts

[18] “BigQuery reddit comments”
https://bigquery.cloud.google.com/dataset/fh-bigquery:reddit_comments?tab=schema

[19] ”Newspaper python library”
http://newspaper.readthedocs.io/en/latest/index.html#

[20] ”Thought vector from google”
https://www.theguardian.com/science/2015/may/21/google-a-step-closer-to-developing-machines-with-human-like-intelligence

[21] ”Google recommendations for training our own data for seq2seq”
https://google.github.io/seq2seq/data/

[22] ”Effective Approaches to Attention-based Neural Machine Translation”
https://arxiv.org/pdf/1508.04025.pdf