MSc in Business Analytics

Full-time Program – 3rd Quarter

Academic Year 2018-2019

BIG DATA CONTENT ANALYTICS

ASSIGNMENT

Dimitris Ntagiantis (F2821822)

Dimitris Gkoumas (F2821812)

Department of Management Science and Technology

Contents

Introduction

For the purposes of this assignment we are going to perform Sentiment Analysis on data derived from the Rotten Tomatoes dataset.

The dataset is comprised of tab-separated files with phrases from the Rotten Tomatoes dataset. The train/test split has been preserved for the purposes of benchmarking, but the sentences have been shuffled from their original order. Each Sentence has been parsed into many phrases by the Stanford parser and each phrase has a PhraseId. Similarly, each sentence has a SentenceId. Phrases that are repeated (such as short/common words) are only included once in the data through a data-cleaning procedure that is definied at a later stage.

The dataset contains two main sub-datasets:

- train.tsv contains the phrases and their associated sentiment labels. We have additionally provided a SentenceId so that you can track which phrases belong to a single sentence.
- test.tsv contains just phrases. Our goal for this assignment is to assign a sentiment label to each phrase.

The sentiment labels are:

- 0 - negative
- 1 - somewhat negative
- 2 - neutral
- 3 - somewhat positive
- 4 – positive

The rest of the report is separated into two parts, the Business and the Technical one.

For the Business Part, we explain what is sentiment analysis, we demonstrate the business value one can derive by performing sentiment analysis and we highlight the usefulness of the findings of a separate analysis in many aspects of business, politics and social actions.

For the Technical Part, we present how we managed to sentiment users based on their reviews obtained from the Rotten Tomatoes dataset. Moreover, we used Python and particularly Keras framework to build a Recurrent Neural Network (RNN), designed for Natural Language Processing (NLP).

We explain in depth the model's architecture, parameters and fit in the data and we examine its performance based on the behavior of its Learning Curves both in train and test datasets.

Business part

What is Sentiment Analysis?

Sentiment Analysis (aka 'Opinion Mining') is the process of identifying and classifying a piece of text to the tone conveyed by it. It is a process that computationally determines whether a piece of writing is positive, negative or neutral or in other words it determines the polarity of the person who wrote that text.

We should highlight that sentiment analysis is not just a trend. On the contrary it is widely used to predict financial performance, understand customers' perception and behavior or even provide early warnings!

However, it is obvious that the accuracy of the polarity's classification can never be 100% since, for example, a machine does not understand sarcasm. But that does not mean that it is not useful.

According to a recent study, people do not agree 60-65% of the time. That means that even if the machine's accuracy is not perfect, it will still be more accurate than human analysis or even when corpus is huge, manual analysis is not feasible leading as to the solution of sentiment analysis.

Why sentiment analysis?

Because it has many fields to apply:

- Business: In marketing field, companies use sentiment analysis to develop their strategies, to understand customers' feelings towards products or brands, to understand how people respond to their campaigns or product launches and to get feedback on the reason consumers do not buy some products. Additionally, it is widely used for Reputation management - or you could also call it brand monitoring, to communicate with customers and subsequently offer better customer support services and of course for competitor monitoring.
- Politics: In political field, sentiment analysis is used to keep track pf political views, to detect consistency/inconsistency between statements and actions at a government level. It can also be used to predict election results globally; it was also applied in the last Greek elections (7/7/19) with significant success.
- Public and Social Actions: Sentiment Analysis also is used to monitor and analyze social phenomena, for spotting potentially dangerous situations and determine the general mood of the blogsphere.

It is obvious that the applications of sentiment analysis in business cannot be overlooked. Sentiment analysis in business can prove a major breakthrough for the complete brand revitalization. The key to running a successful business with the sentiments data is the ability to exploit the unstructured data for actionable insights.

Machine learning models, which largely depend on the manually created features before classification, have served this purpose fine for the past few years. However, deep learning is a better choice as it:

- Automatically extracts the relevant features
- Helps to scrape off the redundant features
- Rules out the efforts of manually crafting the features

Having insights-rich information eliminates the guesswork and execution of timely decisions. With the sentiment data about your established and the new products, it's easier to estimate the customer retention rate. Based on the reviews generated through sentiment analysis in business, you can always adjust to the present market situation and satisfy your customers in a better way. Overall, you can make immediate decisions with automated insights. Business intelligence is all about staying dynamic throughout. Having the sentiments data gives that liberty. If a big idea is developed, we can test it before bringing life to it. This is known as concept testing. Whether it is a new product, campaign or a new logo, just put it to concept testing and analyze the sentiments attached to it.

Furthermore, Sentiment analysis in businesses can be very helpful in predicting the customer trends. Once we get acquainted with the current customer trends, strategies can easily be developed to capitalize on them and eventually, gain a leading edge in the competition.

Owing to the internet savvy era, this experience becomes the text of their social posting and online feedback. The tone and temperament of this data can be detected and then categorized according to the sentiments attached. This helps to know what is being properly implemented with regards to products, services and customer support and what needs improvement.

Therefore, we are able to develop more appealing branding techniques and marketing strategies to switch from torpid to terrific brand status. Sentiment analysis in business can majorly help us to make a quick transition.

To sum up, It's estimated that 80% of the world's data is unstructured and not organized in a pre-defined manner. Most of this comes from text data, like emails, support tickets, chats, social media, surveys, articles, and documents. These texts are usually difficult, time-consuming and expensive to analyze, understand, and sort through.

Sentiment analysis systems allows companies to make sense of this sea of unstructured text by automating business processes, getting actionable insights, and saving hours of manual data processing, in other words, by making teams more efficient.

Advantages of Sentiment Analysis:

- Scalability: There's just too much data to process manually. Sentiment analysis allows to process data at scale in an efficient and cost-effective way.
- Real-time analysis: We can use sentiment analysis to identify critical information that allows situational awareness during specific scenarios in real-time since a sentiment analysis system can help us immediately identify these kinds of situations and take action.
- Consistent criteria: Humans don't observe clear criteria for evaluating the sentiment of a piece of text. It's estimated that different people only agree around 60-65% of the times when judging the sentiment for a particular piece of text. It's a subjective task which is heavily influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all of their data. This helps to reduce errors and improve data consistency.

Sentiment Analysis Challenges:

- Subjectivity and Tone: The detection of subjective and objective texts is just as important as analyzing their tone. In fact, so called *objective* texts do not contain explicit sentiments. All *predicates* (adjectives, verbs, and some nouns) should not be treated the same with respect to how they create sentiment.
- Context and Polarity: All utterances are uttered at some point in time, in some place, by and to some people, you get the point. All utterances are uttered in context. Analyzing sentiment without context gets pretty difficult. However, machines cannot learn about contexts if they are not mentioned explicitly. One of the problems that arise from context is changes in polarity.
- Irony and Sarcasm: Differences between literal and intended meaning (i.e. *irony*) and the more insulting or radicalizing version of irony (i.e. *sarcasm*) usually change positive sentiment into negative whereas negative or neutral sentiment might be changed to positive. However, detecting irony or sarcasm takes a good deal of analysis of the context in which the texts are produced and, therefore, are really difficult to detect automatically.
- Emojis: There are two types of emojis according to Guibon et al.. *Western emojis* (e.g. :D) are encoded in only one character or in a combination of a couple of them whereas *Eastern emojis* (e.g. ¯_(ツ)_/¯) are a longer combination of characters of a vertical nature. Particularly in tweets or reviews, emojis play a role in the sentiment of texts. Sentiment analysis performed over tweets or reviews requires special attention to character-level as well as word-level.
- Defining Neutral: Defining what we mean by *neutral* is another challenge to tackle in order to perform accurate sentiment analysis. As in all classification problems, defining

your categories -and, in this case, the *neutral* tag- is one of the most important parts of the problem.

Sentiment Analysis in Movies & TV-Shows (our case).

Not only do brands have a wealth of information available on social media, but they also can look more broadly across the internet to see how people are talking about them online. Instead of focusing on specific social media platforms such as Facebook and Twitter, we can target mentions in places like news, blogs, and forums –again, looking at not just the volume of mentions, but also the quality of those mentions.

How Sentiment Analysis can be used in our case (Rotten Tomatoes):

- Analyze news articles, blog posts, forum discussions, and other texts on the internet over a period of time to see sentiment of a particular audience.
- Automatically categorize urgency of all online mentions to your brand via sentiment analysis.
- Automatically alert designated team members of online mentions that concern their area of work.
- Automate any or all of these processes.
- Better understand a brand (movie/TV show producers) online presence by getting all kinds of interesting insights and analytics.

We can understand that Sentiment Analysis is useful in brand monitoring because it provides us with the following benefits:

- Understand how your brand reputation evolves over time.
- Research your competition and understand how their reputation also evolves over time.
- Identify potential PR crises and know to take immediate action. Again, prioritize what fires need to be put out immediately and what mentions can wait.
- Tune into a specific point in time. Again, maybe we want to look at just press mentions on the day of your IPO filing, or a new movie/TV Show launch. Sentiment analysis lets us do that.

Additionally, using sentiment analysis (and machine/deep learning), we can automatically monitor all chatter around one brand and detect this type of potentially-explosive scenario while there is still time to defuse it.

Technical Part

About the dataset

The dataset is comprised of tab-separated files with phrases from the Rotten Tomatoes dataset. The train/test split has been preserved for the purposes of benchmarking, but the sentences have been shuffled from their original order. Each Sentence has been parsed into many phrases by the Stanford parser and each phrase has a PhraseId. Similarly, each sentence has a SentenceId. Phrases that are repeated (such as short/common words) are only included once in the data through a data-cleaning procedure that is definied at a later stage.

The dataset contains two main sub-datasets:

- train.tsv contains the phrases and their associated sentiment labels. We have additionally provided a SentenceId so that you can track which phrases belong to a single sentence.

- test.tsv contains just phrases. Our goal for this assignment is to assign a sentiment label to each phrase.

The sentiment labels are:

- 0 - negative
- 1 - somewhat negative
- 2 - neutral
- 3 - somewhat positive
- 4 – positive

Analysis Methodology explanation

Our analysis will be based on very popular Python packages and libraries. More specifically, we are going to use 'Pandas' a package for high-performance, easy-to-use data structures and data analysis accompanied by the 'NumPy' which is a fundamental package for scientific computing with Python.

Furthermore, we are going to use the Natural Language Toolkit (NTLK) which is a collection of libraries and programs for symbolic and statistical natural processing for English written texts in Python.

Last but not least, we should highlight that in order to make our analysis performance more efficient, we are going to use the TQDM library. TQDM is a progress bar library with good support for nested loops and Jupiter/Python notebooks.

Regarding the deep learning framework, we are going to use Keras. Keras is a deep learning framework that actually under the hood uses other deep learning frameworks in order to expose a simple to use and work with high-level API. In our case, Keras will use the Tensorflow – Google's deep learning library.

The integration with the various backends is seamless and comes in two flavors: sequential or functional. Just to ways of thinking about building models. The resulting models are perfectly equivalent.

We're going to use the sequential one, to develop various types of models for Natural Language Processing.

- (Dense) Deep Neural Network – The NN classic model – uses the BOW model
- Convolutional Network – build a network using 1D Conv Layers – uses word vectors
- Recurrent Networks – LSTM Network – Long Short-Term Memory – uses word vectors
- Transfer learning for NLP – uses word vectors

Loading and Cleaning our data

After importing all the necessary libraries and packages we described above, we proceed by importing the two datasets (train & test tsv files) to two pandas data frames with the same name.

The train dataset, which is going to be used to train our model, contains 156.060 rows which are reviews and 4 columns that offer us information about the reviews' characteristics along with the exact phrase and has the following form:

	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage ...	1
1	2	1	A series of escapades demonstrating the adage ...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2

Similarly, the test dataset (in which we are going to test the predictive performance of our model in terms of sentiment classification) follows the same shape as the train dataset with less rows (66.292) and 3 columns, the same with the train dataset except the 'Sentiment' column which we are going to predict through our RNN model.

The test dataset has the following form:

	PhraseId	SentenceId	Phrase
0	156061	8545	An intermittently pleasing but mostly routine ...
1	156062	8545	An intermittently pleasing but mostly routine ...
2	156063	8545	An
3	156064	8545	intermittently pleasing but mostly routine effort
4	156065	8545	intermittently pleasing but mostly routine

The next step is to create a function for cleaning the reviews, tokenize and lemmatize them. This function will take each phrase iteratively and it will do the followings:

1. remove html content

2. remove non-alphabetic characters

3. tokenize the sentences

4. lemmatize each word to its lemma

5. return the result in the list named reviews.

Dataset's Pre-processing

Before we start building our model, we have to perform some extra steps to transform our data in a form suitable for our RNN model.

These steps are:

1. Collect the dependent values and convert to one-hot encoded output using to_categorical

2. Split into train and validation datasets

3. Getting the no of unique words and max length of a review available in the list of cleaned reviews

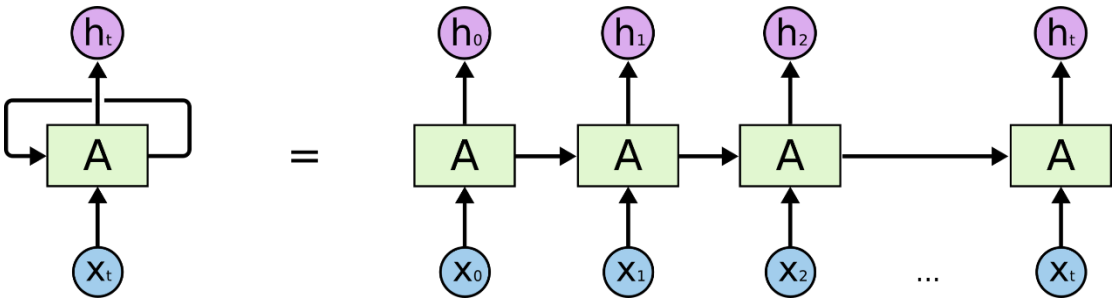
4. Tokenizing and converting to sequences

5. Perform padding to equalize the lengths of all input reviews since LSTM networks needs all inputs to be same length

6. Define early-stopping threshold to prevent overfitting

Model's Architecture and Parameters

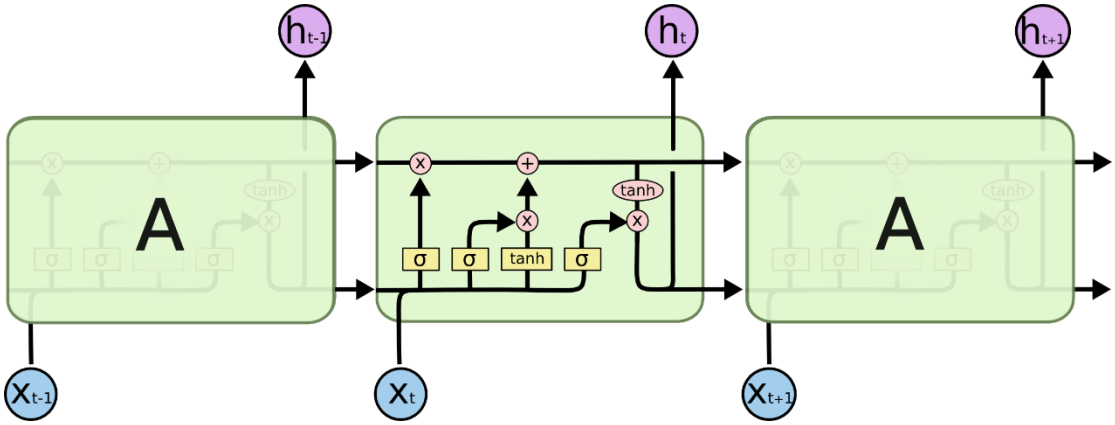
In order to train the model, we are going to use a type of Recurrent Neural Network (RNN), known as LSTM (Long Short-Term Memory). The main advantage of this network is that it is able to remember the sequence of past data i.e. words in our case in order to make a decision on the sentiment of the word.



An RNN Network (source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

As seen in the above picture it is basically a sequence of copies of the cells, where output of each cell is forwarded as input to the next. LSTM network are essentially the same but each cell architecture is a bit more complex.

This architecture is specially designed to work on sequence data since it fits perfectly for many NLP tasks like tagging and text classification. More specifically, it treats the text as a sequence rather than a bag of words or as engrams. This feature, allows each cell to decide which of the past information to remember and the ones to forget, a great advantage in terms of performance's speed and memory's use.



A LSTM Cell (source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Then, we are going to create the network using Keras. Keras is built on tensorflow and can be used to build most types of deep learning models. We are going to specify the layers of the model as below.

In order to estimate the parameters such as dropout, no of cells etc..., we have performed a grid search with different parameter values and chose the parameters with best performance.

As far as the model's parameters are concerned, we should focus on three main parameters and explain the values/types selected.

These parameters are the following:

- Activation Function: We have used ReLU as the activation function. ReLU is a non-linear activation function, which helps complex relationships in the data to be captured by the model.
- Optimizer: We use adam optimizer, which is an adaptive learning rate optimizer.
- Loss function: We will train a network to output a probability over the classes using Cross-Entropy loss, also called Softmax Loss. It is very useful for multi-class classification.

Model's Performance based on Learning Curve

We are going to visualize the behavior of the learning curve both for training and validation datasets to check the performance of our model in an understandable way.

A learning curve is a plot of model learning performance over experience or time or more simply, it is a line plot of learning (y-axis) over experience (x-axis).

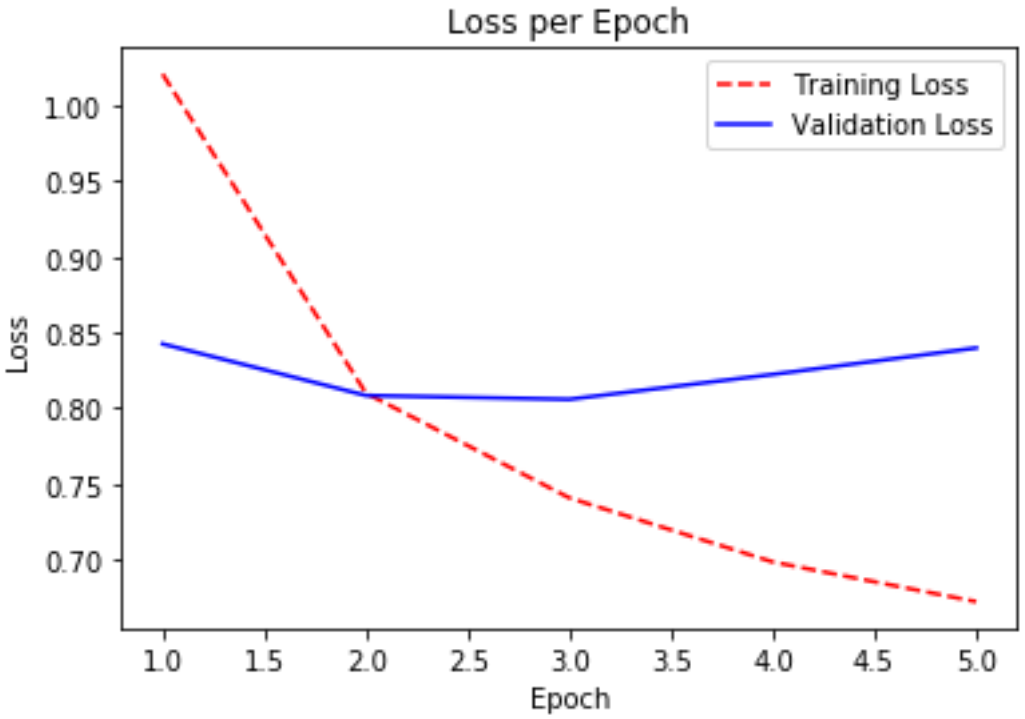
Learning curves are a widely used diagnostic tool in machine and deep learning for algorithms that learn from a training dataset incrementally. The model can be evaluated on the training

dataset and on a holdout validation dataset after each update during training and plots of the measured performance can created to show learning curves.

Reviewing learning curves of models during training can be used to diagnose problems with learning, such as an underfit or overfit model, as well as whether the training and validation datasets are suitably representative.

During the training of a machine learning model, the current state of the model at each step of the training algorithm can be evaluated. It can be evaluated on the training dataset to give an idea of how well the model is "*learning*." It can also be evaluated on a hold-out validation dataset that is not part of the training dataset. Evaluation on the validation dataset gives an idea of how well the model is "*generalizing*."

- **Train Learning Curve**: Learning curve calculated from the training dataset that gives an idea of how well the model is learning.
- **Validation Learning Curve**: Learning curve calculated from a hold-out validation dataset that gives an idea of how well the model is generalizing.



We observe that here the learning curve is not ideal, it should be smoother as it decreases.

Model's average accuracy is equal to 67% approximately, which can be considered a pretty good fit. A good fit is the goal of the learning algorithm and exists between an overfit and underfit model.

A good fit is identified by a training and validation loss that decreases to a point of stability with a minimal gap between the two final loss values.

The loss of the model will almost always be lower on the training dataset than the validation dataset. This means that we should expect some gap between the train and validation loss learning curves. This gap is referred to as the "generalization gap."

A plot of learning curves shows a good fit if:

- The plot of training loss decreases to a point of stability.
- The plot of validation loss decreases to a point of stability and has a small gap with the training loss.

Apparently, the behavior we just described is present in our plot so we can reasonably consider our model's fit to be good.

Last but not least, we are confident to proceed to the last step of our Sentiment Analysis which is to predict the sentiment of the Rotten Tomatoes users' reviews and classify them based on the RNN model we built for the purposed of this assignment.

Conclusion and comments

Using the model, we built we can extract business value in several ways. The most major ones are the following:

- Analyze news articles, blog posts, forum discussions, and other texts on the internet over a period of time to see sentiment of a particular audience.
- Automatically categorize urgency of all online mentions to your brand via sentiment analysis.
- Automatically alert designated team members of online mentions that concern their area of work.
- Automate any or all of these processes.
- Better understand a brand (movie/TV show producers) online presence by getting all kinds of interesting insights and analytics.

As we mentioned before, sentiment analysis is not just a trend. On the contrary it is widely used to predict financial performance, understand customers' perception and behavior or even provide early warnings. This was the motivation to perform this analysis and present the results in terms of our assignment.