National College of Ireland

Movie Spoilers Classification Over Online Commentary, Using Bi-LSTM Model With Pre-trained GloVe Embeddings

MSc Research Project
Data Analytics

Anyelo Lindo
Student ID: x18170412

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anyelo Lindo
Student ID:	x18170412
Programme:	Data Analytics
Year:	2019
Module:	MSc Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	25/05/2020
Project Title:	Movie Spoilers Classification Over Online Commentary, Using Bi-LSTM Model With Pre-trained GloVe Embeddings
Word Count:	6230
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	25th May 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Movie Spoilers Classification Over Online Commentary, Using Bi-LSTM Model With Pre-trained GloVe Embeddings

Anyelo Lindo
x1817012

Abstract

In the past few decades, society and it's ways of living has been reshaped in order to adapt to the ongoing technological breakthroughs. Perhaps, what was not use to be much of an issue in the past, now can be a problem. Movie spoilers in particular, have now become a matter of concern for cinema fanatics and the film industry itself. Moreover, freedom of speech on virtual communities where film fanatics meet, can be a source of dread as user-generated data comes to be challenging for moderation. Due to the fact that these commentary reviews may contain revelatory information associated to movies plot, that could ruin real-time cinema experience for thousands of movie-goers who enjoy attending films screening. Ergo, impacting the cinematic industry revenues as well. In this wise, we proposed a supervised deep learning model that will serve as foundation for future work on this field. Using Bidirectional Long Short-Term Memory(Bi-LSTM) with pre-trained Global Vectors (GloVe) to improve the accuracy in the text classification, as well as the training speed; so to deal with spoilers over online commentaries about movie reviews. Additionally, for testing purposes we used two different well-known methods to extract features from text for modeling: Bag Of Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF); so as to build four extra classifiers: Support Vector Machine (SVM), Logistic Regression (LR), Bernoulli Naive Bayes (NB) and Random Forest (RF). Results obtained were satisfactory and speak on behalf of our proposed solution. On the grounds that our model not only achieved good skill to discriminate between classes compared to the rest of the classifiers, but also completed the training in a fairly short time.

Keywords: *Bi-LSTM, Deep Learning, GloVe, NLP, Spoilers, Supervised, Text Classification, Word Embeddings.*

1 Introduction

Within the past few decades, humanity has started to change in a constant pace. Some might see it as adaptation, some others as progress. In either case, many aspects of our civilisation like: health, education, well-being, communications, etc. Tend to cross paths between each other at some point. In all honesty, sometimes this inborn necessity is not entirely achieved in a natural way. In contrast, it is being hasten by external phenomenons such as the one we've been witnessing since the late 20^{th} and the beginning of the 21^{st} century, up to the present time with a plethora of technology. More importantly, it's accelerated expansion and evolution, trans-disciplinary speaking, plays one of the main roles if not the main, in the contemporary society. At such extend that it has now reshaped the way of individuals communicate and interact between each other. Film industry for instance, as

a combination of Art forms, also appeals to serve these social interactions aiming different purposes by targeting divergent niches in pursuance of creating a profitable business, while delivering entertainment to a whole new generation of fans. Who have now started to take advantage of online tools to build like-minded communities where they can interact, consume, share, learn and find all type of information related to any specific movie franchise, genre or even crew performances. However, this novelty way of synergy between these virtual communities comes with a downside. Whenever these movie followers come across with online commentaries that contain pieces of information that are relevant to the film plot. These type of content is well-known as spoilers. This phenomenon[1] is now being widely observed as it has grown to become a major issue in the modernized society[2], not only because spoilers can steal the joy of experiencing a film at first hand from these millions of new cinema cult of devotees. But also because it does affect the trade of the Film Industry [3] itself. Which can be reflected in the movie box office during the screen premier and at some time in the future, the marketing machinery behind a cinematic production and more future projects funding. With regards to the movie spoilers significance in the present-day and the explosion of user-generated content on Web sites. We can relate on Text Mining as a sub-field of Data Mining and it's different methods to face these issues. Therefore, the aim of this project seeks to built a novel supervised Machine Learning (ML) model with a Natural Language Processing (NLP) approach. Using Bidirectional Long Short-Term Memory (Bi-LSTM) and pre-trained Global Vectors (GloVe) to improve the accuracy in the text classification for online commentaries on movie reviews. Thus, improving the accuracy of detecting spoilers within the text to counterbalance the spoiler-phobia phenomenon that spreads through all these modern online communities. At the same time that pursues providing a trustworthy work that serves as foundation for future efforts against spoilers deal-breaker for global culture and film-makers industry.

2 Machine Learning and Natural Language Processing (NLP)

Back in the old days, we used to live in an world where we did not break everything down into binary code to work. As a matter of fact Spoilers have always been out there, and the paradox of whether they do enhance the experience of watching a film or not, has been approached from multidisciplinary angles. Such as the psychological context (Johnson and Rosenbaum; 2015) where several variables have been taken into account, sharply showing that Spoilers do actually affect audience's enjoyment. In contrast, with what is commonly assumed. Spoilers might upturn the experience by leveraging the fluency of the narrative (Leavitt and Christenfeld; 2014). Either way, Spoilers have always found a way to wreck one's movie plots joy experience or the business backstage itself; as the case may be that either one's used to hear up-close some else talking about a film narrative nor as simple as back then they were not use to be considered a phenomenon just because they used to take place once in a blue moon, thus not impacting significantly the Film industry profits. Present-day with the technology out-burst, our society has now learned new means to deal with the upsurge of movie Spoilers issue. Natural Language Processing (NLP) for instance, as a sub-field of the field of Artificial Intelligence (AI), offers a variety of tasks which can provide the possibility of capturing the sentiment of the vast social media languages by analysing key information that helps to identify the polarity on the online commentary which could lead to built decision support systems (Fersini et al.; 2016). Furthermore, the performance of these tasks over on-

[1] Spoiler-phobia: https://www.washingtonpost.com/business/2019/05/22/spoiler-fight-hollywood-has-its-free-speech-moment/

[2] Spoilers:https://www.thenational.ae/arts-culture/film/how-did-spoilers-become-bigger-news-than-the-films-they-re-spoiling-1.853450

[3] U.S. Film industry revenue:https://deadline.com/2018/07/film-industry-revenue-2017-ibisworld-report-gloomy-box-office-1202425692/

line user-generated text could be improved applying Word Embedding[4] techniques combined with Gaussian models to attempt outlining the continuous distribution for short commentary's, in order to catch semantic connections between words (Ma et al.; 2015). Since words themselves are the root of every text classification process, novelty models using Naive Bayes and Bag-of-Embeddings probabilities attributes have been held. These simple models can achieve competitive accuracy's scores against more complex ones, because they infer the probabilities of class-sensitive of the bag-of-embeddings vectors with the context of the sentences (Jin et al.; 2013). Nonetheless, dealing with raw text becomes a challenging task as predictions accuracy can fall in the noisy domain with ease. Therefore innovative research's based on mathematical approaches using Non-negative Matrix Factorisation (TONMF) algorithm (Kannan et al.; 2017), have shown better performance dealing with outliers by using non-negative matrix factorization (NMF), because they are able to instinctively recognise inconsistencies with the usage of low rank approximations. On the other hand, commonly used algorithm's like Kernel PCA (Principal Component analysis), Distant Based Method, Distant K-based Method, Simple Statistical Method, Kernel-Based Novelty Detection Method and One-class Support Vector Machine (SVM) method; have also shown fair effectiveness for outliers detection on Machine Learning models (Escalante; 2014). However, there is no such thing as perfect accuracy in a text corpora model. Differently, Word Embeddings as numerical representation of each word and their similarities in a corpus, has become a powerful tool for Machine Learning modeling. Proving to perform greatly over text classification tasks, along side with Convolutional Neural Networks (CNN) and pre-trained vectors (Amit Mandelbaum; 2016). Although, these type of techniques tend to leave out the sentiments information of documents during a Natural Language Processing model, which can lead to deliver poor accuracy scores whereas an small size of text corpus is being used to train. Yet some remarkable Word Embedding based work like Global Vectors (GloVe) and Word Vectors (Word2Vec[5]) have been developed to address this issue, so that the accuracy during a sentiment analysis notably increases (Rezaeinia et al.; 2017) as result of being trained in advance over a large text corpus (millions of words).

2.1 Machine Learning Classifiers for Text Classification

Classifiers are defined as the algorithms which are built for the mere purpose of learning from a training process using input data, in order to classify new findings related to a given class [6]. Moreover, there exist several types of algorithms in Machine Learning that are implemented according to the classification goal. As the case of text corpus classification, some classifiers might deliver higher performance than others. Binary Logistic Regression classifier as a class of regression for example, usually applies Bag of Words (BOW) as method to represent this binary categorisation. Previous work has improved the efficiency compared to well-known classifiers, by learning from the tokenization [7] process itself, using Logistic Regression combined with Gradient Ascent technique of all N-grams (Ifrim and Weikum; 2008). On the contrary, Random Forest can handle high dimensional text data containing diverse classes. Where the performance of a classification problem can be improved by adding new feature weighting and a tree selection method (Xu; 2012). Naive Bayes classifier is based on the Bayes theorem that assumes independence among predictors. Such assumption could be incorrect for some cases, therefor previous research have proposed an ad-hoc method to incorporate document length feature (Eyheramendy; 2019). Support Vector Machine for instance, can be used for classification and regression tasks. Although, is broadly used for common topic text classification problems within very interesting approaches like including feature se-

[4]Word Embedding: `https://machinelearningmastery.com/what-are-word-embeddings/`

[5]Word2Vec: `https://en.wikipedia.org/wiki/Word2vec`

[6]Classifier:`https://en.wikipedia.org/wiki/Statistical_classification`

[7]Tokenization: `https://learning.oreilly.com/library/view/python-machine-learning/9781786464477/ch06s02.html`

lection method of Chi-Square Difference between positive and negative categories(CDPNC), using Document Frequency and Chi-Square in pursuance of performance leverage (Luo; 2016). Recently, Machine Learning models using Recurrent Neural Networks (RNN) have become risen in interest for data scientist's, because it's widely applicability on NLP problems. As well as for providing feed-forward learning through several available architectures of RNN composed of layers, which can improve the performance depending on the exploration of different task, using common features (Learning et al.; 2011). Overall, classifiers performance will be conditioned by several factors such as type of problem, size of the training data, expected outcome, computing time, etc. To give an idea about there is no such thing as perfect classifier, a comparison work between most common algorithms (Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression methods for multi-class text classification) (*Comparison of Naïve Bayes , Random Forest , Decision Tree , Support Vector Machines , and Logistic Regression Classifiers for Text Reviews Classification*; 2017), might show that Logistic Regression multi-class out-performed other competitors.Even though is not the most preferred choice for text classification analysis[8].

2.2 Related Work

Movie spoilers issue is thriving in times like these where the amount of user-generated data spread quickly and has become almost unstop-able, owing to technological actuality. Which provides easy-free will interaction between movie fanatics expressing individual opinion about a film on the Internet, in the form of comments. Thus, past research projects have also explored miscellaneous methods such as topic models based on Latent Dirichlet Allocation (LDA) using linguistic dependencies information (Guo and Ramakrishnan; 2010), to face spoilers phenomenon over online commentary. In particular, Machine Learning end-to-end neural network architecture that combines Hierarchical Attention Network (HAN)[9] and predictive Binary class, has shown fine results trying to detect sensitive plot twists sentences from review documents (Nakashole; 2019). Regarding to online comment tendency, sentiment analysis approaches increased favouritism over the past few years due to the necessity of identifying polarity levels from users responses. Prior research suggests that building cooperative classifiers which are well-known for achieving good results on a particular task, might show better outcome (Catal and Nangir; 2017). Additionally, incorporating novel improved Word2Vec model with Part-of-Speech tagging techniques, into the sentiment analysis model, have proven to help boosting the accuracy of the prediction (Mahdi et al.; 2019). Thus assuring that Machine Learning tasks analysis over text, can relay on Word2Vec because it is able to capture semantic relationships between words to provide much better results (Zhang et al.; 2015). What is more, Word Embeddings work also great on topic models (Liu et al.; 2015) by attaching topics to every word, respectively. So that the Topical Word Embedding (TWE) produces a flexible framework where every single word within the text corpus is allowed to have different numerical representation, in line with the topic that word is related to. In particular, GloVe for word representations (Pennington et al.; 2014), along side Word2Vec, is one of the most used algorithms for embedding purposes. GloVe is an unsupervised log-bilinear regression model that surpasses similar models, because maps words into a vector space where the distance between words depend on semantic similitude's. Case in point is RNN recurrent architecture, offers distributed representation of words by breaking documents into tokens and place them in vectors. Aforesaid matrix consists of two dimensions: Time-step and Features. Conversely many models only take into count one-dimension (either max pooling[10] or attention based[11]) which might ruin the structure of the feature representation. This has been dis-

[8] Text analysis classifier: `https://monkeylearn.com/text-analysis/`

[9] HAN: `https://medium.com/analytics-vidhya/hierarchical-attention-networks-d220318cf87e`

[10] Max Pooling: `https://computersciencewiki.org/index.php/Max-pooling_/_Pooling`

[11] Attention-based: `https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129`

cussed in previous works (Zhou and Qi; 2016) where researches have explored the integration of features, on both dimensions so that the samples obtained from the matrix, could include more substantial information. In a like manner Bidirectional Long Short Term Memory (BiLSTM) is a class of RNN used for deep learning (sequential) tasks, where the learning algorithm spreads the input sequence forward and backward, through hidden layers. BiLSTM models have been broadly applied in different fields with text related tasks, like Medical named entity recognition with Conditional Random Fields (Bi-LSTM-CRF) (Xu et al.; 2018). Where Machine Learning can catch character based representations, for the sake of identifying specific health realm terminology in an efficient fashion. Bi-LSTM is not only suitable for named entity recognition scene but also to find the main topics in a collection of documents without having the necessity of manually performing any other interations (Basaldella et al.; 2018). By the same token, Bi-LSTM have also demonstrated achieving good results over movie genre classification problems. When the training data is limited down only to plot summaries. The reason behind this strategy is that them summaries contain enough information to capture genre details. What is more, the sentences are divided individually and passed through the Bi-LSTM network to boost it's training time, rather than using whole plot text (Ertugrul; 2018).

3 Implementation and Methodology

Researches in the Machine Learning field have always relayed their work on data mining methodologies, because these specialised techniques provide adequate guidance to establish the most suitable road-map that meets project proposal. Namely, the Cross-Industry Standard Process for Data Mining (CRISP-DM) is the methodology chosen to follow in this research, because poses appropriate approaches that were needed to perform all activities during the life-cycle of this research project. CRISP-DM present a total of six phases: *Business Understanding*[3.1], *Data Understanding*[3.2], *Data Preparation*[3.3], *Modeling*[5], *Evaluation and Deployment*[6]. These phases are flexible in terms of not posing a mandatory order between them, because all tasks can be performed at any order, and whats more, they offer the possibility of reversing to previous stage's in order to repeat specific activities as the project requires. These phases will be discussed in further details within following sections.

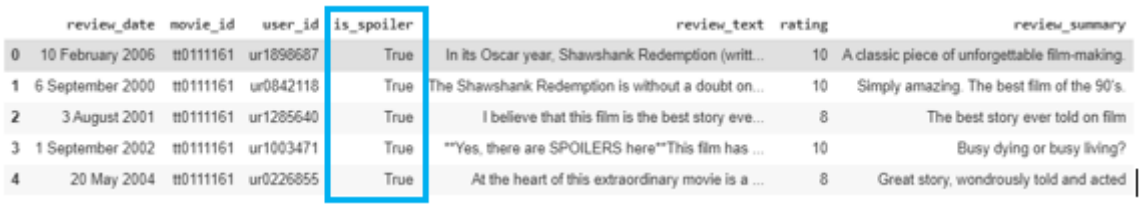
3.1 *Business Understanding*

The first phase focuses on understanding what the business is all about and consequently, what is the objective that this study is expected to achieve. Therefore, the goal of this research project is to answer the novel question: ***Can Movie Spoilers detection be improved, over online reviews using Bi-LSTM with Pre-trained Word Embeddings?***. Thus, by building a Machine Learning model that serves as foundation for future works. Where online communities of movie fanatics and the film industry business itself, can count on a trustworthy appliance that is capable of identifying commentary that reveals important plot twist of a movie.

3.2 *Data Understanding*

This phase covered several steps which included data: collecting, describing, exploring and quality validation. So that future problems like data consistency during the other phases, can be avoided. The data used for this project consisted of two different datasets collected from Kaggle and IMDB sites. Conceding the objective identified for this research during the previous phase, the model result is defined as supervised learning. Where the chosen algorithm possible outcomes need to be known in advanced and the data used to train the model is already labelled with correct answers. In

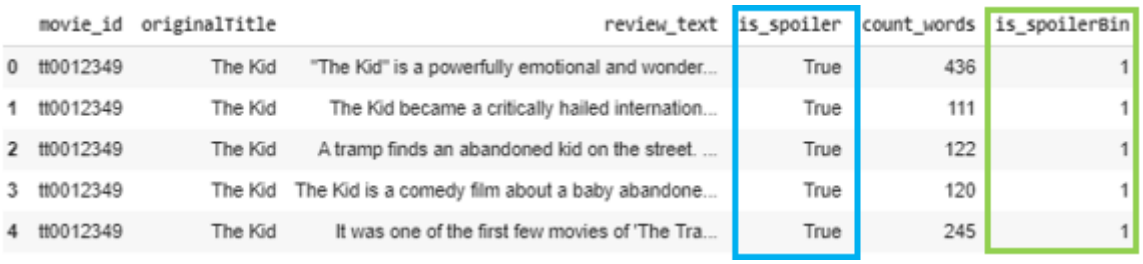
this fashion Kaggle's dataset already contained annotated data classifying whether a commentary review contains spoiler or not.



	review_date	movie_id	user_id	is_spoiler	review_text	rating	review_summary
0	10 February 2006	tt0111161	ur1898687	True	In its Oscar year, Shawshank Redemption (writt...	10	A classic piece of unforgettable film-making.
1	6 September 2000	tt0111161	ur0842118	True	The Shawshank Redemption is without a doubt on...	10	Simply amazing. The best film of the 90's.
2	3 August 2001	tt0111161	ur1285640	True	I believe that this film is the best story eve...	8	The best story ever told on film
3	1 September 2002	tt0111161	ur1003471	True	**Yes, there are SPOILERS here**This film has ...	10	Busy dying or busy living?
4	20 May 2004	tt0111161	ur0226855	True	At the heart of this extraordinary movie is a ...	8	Great story, wondrously told and acted

Figure 1: Kaggle annotated dataset

Kaggle dataset was initially collected from IMDB site with online commentary reviews, and then uploaded to Kaggle with annotated data. These data specifies whether a review contains a spoiler or not. And consist of: 573913 records, 263407 register users, a total of 1572 reviewed movies, 79039 users with at least one spoiler review. Later on, simple pre-processing steps were performed to remove data that is not relevant for our classification model (*user_id, rating* and *review_summary* columns). In parallel with previous steps, preparation for farther tasks needed to be addressed. As Machine Learning models basically learns from numbers, the independent variable in this dataset is the column *is_spoiler*; and is a Dichotomous variable. Which means that there are only two possible answers to the question: **does a movie review contain spoilers?** *True* or *False*. To this end, an additional binary encoded column (called *is_spoilerBin*) representing the independent variable values (*True = 1* or *False = 0*), was created. The overall result of these actions are:



	movie_id	originalTitle	review_text	is_spoiler	count_words	is_spoilerBin
0	tt0012349	The Kid	"The Kid" is a powerfully emotional and wonder...	True	436	1
1	tt0012349	The Kid	The Kid became a critically hailed internation...	True	111	1
2	tt0012349	The Kid	A tramp finds an abandoned kid on the street. ...	True	122	1
3	tt0012349	The Kid	The Kid is a comedy film about a baby abandone...	True	120	1
4	tt0012349	The Kid	It was one of the first few movies of 'The Tra...	True	245	1

Figure 2: Kaggle data after initial pre-processing

Table 1: Kaggle's data overview

Overall Description	**Total count**
Dataset shape	(573913 rows, 7 columns)
Null values	0
Mean number of words per review	259.48 words
Max. number of words within a review	2675 words
Min. number of words within a review	1 word

On the other hand, IMDB dataset was collected from it's site, for the mere purpose of combining it with Kaggle's dataset so that movie's titles could be included for further possible references. Hence, the analysis of this data was not performed at the same level as for Kaggle's. For instance, this dataset initially included information related to diverse title's type like: movies, series, documentaries, shorts, TV episode and so on. Since Kaggle's dataset only includes reviews for movies, to begin with, the performing of filtering actions were executed. So the rule *titletype* is not equal to

movie, was applied. Later simple pre-processing steps were also performed, in order to remove not required data for our model (*titleType, primaryTitle, isAdult, startYear, endYear, runtimeMinutes* and *genres* columns). The overall result of these actions is:

	tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes	genres
0	tt0000001	short	Carmencita	Carmencita	0	1894	\N	1	Documentary,Short
1	tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	0	1892	\N	5	Animation,Short
2	tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892	\N	4	Animation,Comedy,Romance
3	tt0000004	short	Un bon bock	Un bon bock	0	1892	\N	\N	Animation,Short
4	tt0000005	short	Blacksmith Scene	Blacksmith Scene	0	1893	\N	1	Comedy,Short

Figure 3: IMDB dataset

Table 2: IMDB's data overview

Overall Description	**Total count**
Dataset shape	(535541 rows, 2 columns)
Null values	0

Afterwards, since no data was found missing, both datasets were merged into one and so an exploratory data analysis supported by visualisations, was carried out to determine text corpora distribution. Revealing that 148.182 (accounts for 26 % of data) reviews contain spoilers, and 416.856 (accounts for 73 % of data) reviews are spoilers free. Thus evidencing the presence of small class imbalance, which we will deal with it in further phases.

3.3 *Data Preparation*

Considering what we've learnt up to this point about the data distribution. Next step aimed to pre-process the data so that we could turned it into meaningful pieces of information to fed Machine Learning algorithms. That being said, during this stage, normalisation and transformation of the text becomes subject of matter.Before anything else, it was necessary to break text up into individual sequences of characters so-called tokens [12]. Furthermore, tokenize is one of the most essential tasks of the NLP activities, because helps to identify patterns and boundaries between every word in the text corpora. Once tokenization process has finished, following tasks were executed:

- ☐ Emojis and emoticons belong to the raw form of the text that is commonly used among online communities. Although, there are now some researches including these variables in the NLP equation, these kind of user-generated inputs do not generally cope with NLP processes for classification.

- ☐ Removal of contractions(also known as the shortened form of words), punctuation, stop-words (i.e. of, the, a, an, in, etc), tags or line breakers, multiple white spaces between words, numbers, special symbols. Were also removed as these do not add up anything to the model accuracy, but noise.

- ☐ The next step aims to reduce words to a common root, to pave the way for NLP analysis. On that wise, the appliance of Lemmatization was chosen over the Stemming technique, as finding

[12] `https://learning.oreilly.com/library/view/natural-language-processing/9781617294631/kindle_split_012.html`

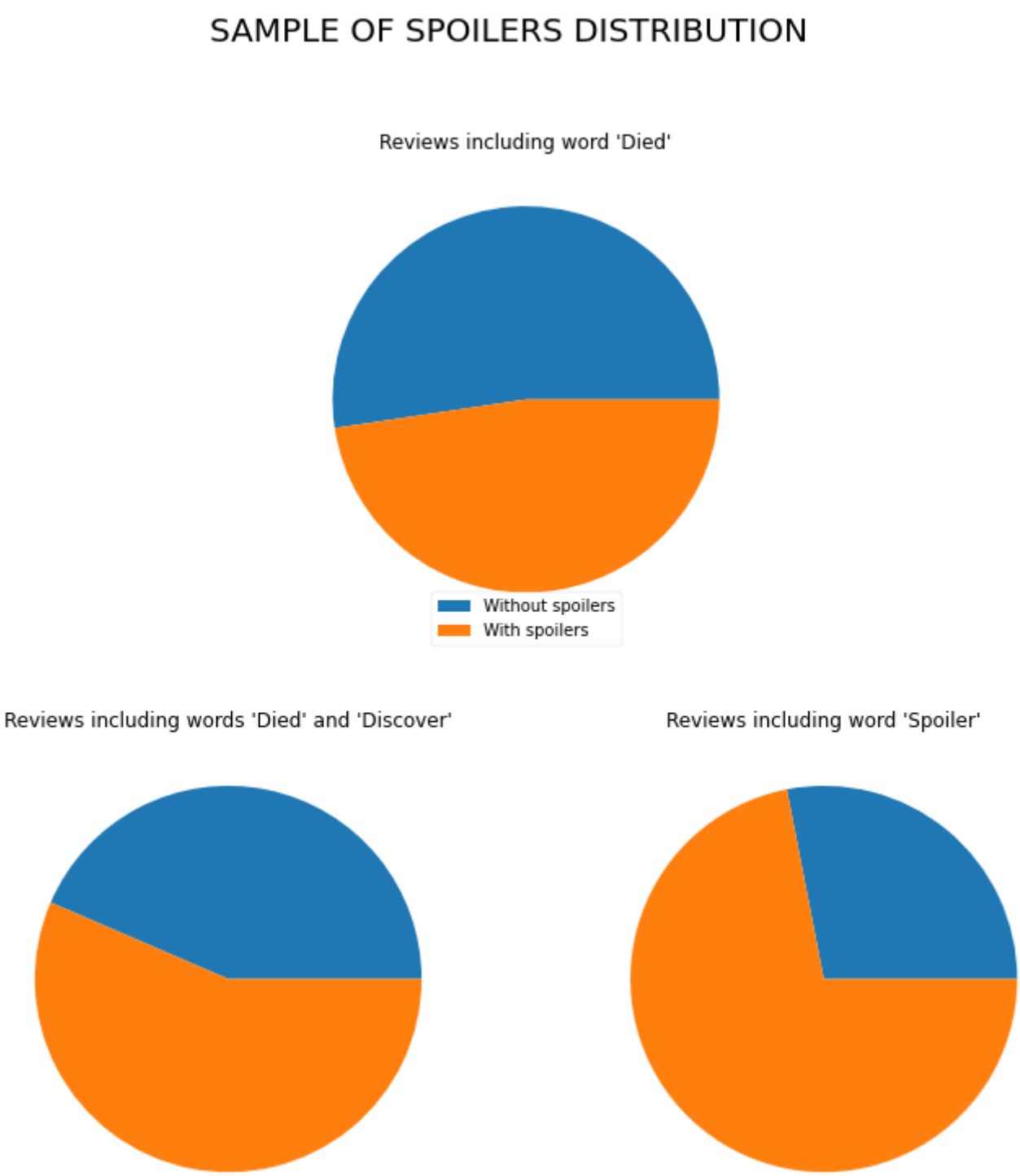


Figure 4: Sample of movie spoilers reviews

it more suitable to our goals. The rationale behind this selection, is because Stemming only cuts off suffixes indiscriminately, perhaps returning non-valid words. Whereas Lemmatization goes beyond and performs a morphological words breakdown to transform them back to their original form, i.e. the word *Studies*, would be returned as: $Stem \longrightarrow$ *Studi* Vs $Lemma \longrightarrow$ *Study*.

In this sense, Lemmatization algorithm was decided to be applied over the remaining text, for better grammar results during the NLP tasks. But first upper cases need to be dealt with, to avoid machine mistaking identical words as if they were two different ones i.e. *Language* is

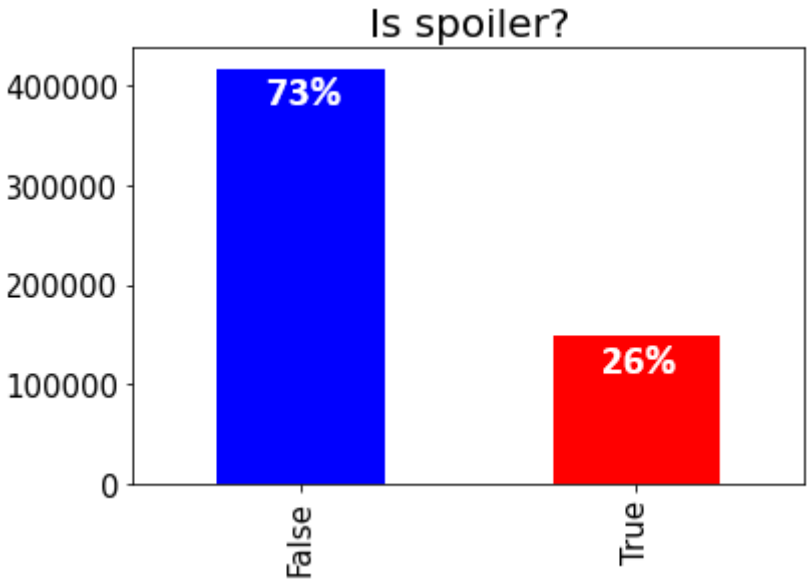


Figure 5: Complete distribution of the spoiler data

the same word as *language*, but from machine angle they are two distinct words. Once upper cases forms are fixed, we can now remove affixes from words to transform them back to their root form. As stated above, this process is known as Lemmatization.

- ☐ Finally, user-generated text data tends to be messy as there is not much, or most likely none control over what users write in these online movie fans communities. So it becomes necessary to try to get fixed as many syntax mistakes as we can before building our models. In this fashion, spelling correction procedure becomes handy. As it will increase the quality of the data for our model, by fixing those words that have made it this far in the form of grammar mistakes.

Above all, text is now normalized and ready to serve our Machine Learning model building. Nonetheless, we must note that even after successfully performing previous pre-processing actions, there's not own manually coded function nor built-in programming language package, that will actually deliver 100% of success rate. Since we are dealing with a large text corpus with millions of words, there will be always more room to enhance this process in order to get rid of unwanted noise, so that we could improve the quality of the data.

There are two additional CRISP-DM phases, that will be cover in further sections of this document(Modeling 5, Evaluation and Deployment 6).

4 Design Specification

The design of our Bi-LSTM with GloVe embeddings model, relates to all CRISP-DM methodology phases explained above. Where the start point is the data collection from the sources: Kaggle and IMDB sites. Subsequently, a cycle stage that includes pre-processing activities was undertaken.Such activities seek to normalize the data into a suitable form which meets the minimum requirements to be fed to GloVe embedding process, for the purpose of being transformed into a vector of representations for words. Later, this transformed data is passed down to our model for training, in order to

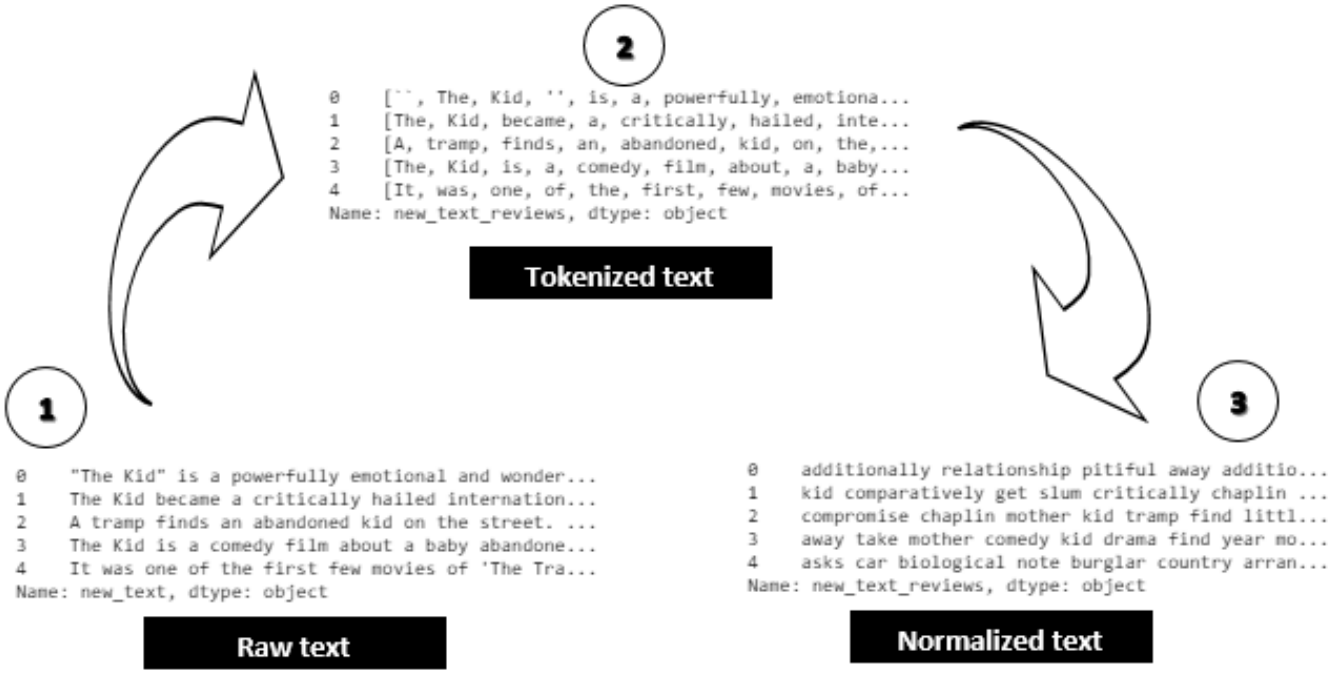


Figure 6: Overview of data preparation process

achieve project goals. Finally, these initial results will be evaluated to decided whether our model Hyper-parameters[13] tuning options, do still have room for performance improvement.

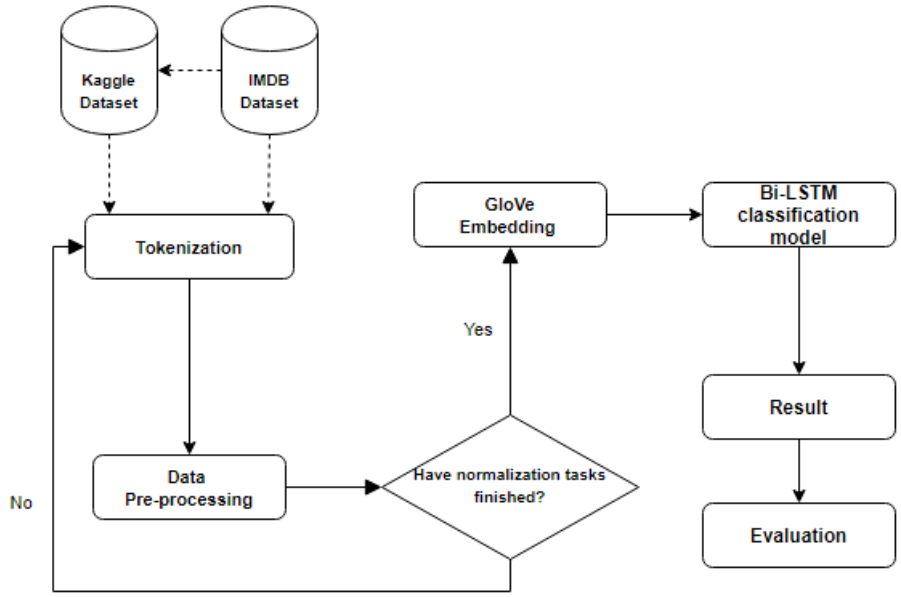


Figure 7: Bi-LSTM with GloVe Embeddings design

[13]Hyper-parameters:https://www.datacamp.com/community/tutorials/parameter-optimization-machine-learning-

5 Modeling

In this section we illustrate a total of five models that were built and trained for binary classification prediction purpose: Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF) and our novel proposal for spoiler classification, Bi-LSTM with pre-trained GloVe embedding model. Moreover, as every Machine Learning model algorithm instead of text, requires having numeric sequences to work with. In this degree and for accuracy comparison purposes, we've selected three different techniques that are well-known for transforming strings into numerical vectors of words. Hence these vectors can act as representation of the data during the training of our Machine Learning algorithms.
Word embeddings is a known terminology for algorithms that are able to map a group of characters, words or sentences to vectors of continuous numbers that serve as data during the training/test phase. In this manner we've chosen to apply following word representation techniques: Global Vectors (GloVe), Term Frequency Inverse Document Frequency (TF-IDF) and Bag Of Words (BOW). In view of representing text documents for BI-LSTM, SVM, LR, NB and RF models.
The implementation of this research was achieved taking advantage of the simplicity and readability of *Python 3.7* programming language. Boosting the computational power using *Jupyter Notebook* with GPU's over Google Colab free cloud service.

5.1 *TF-IDF for Support Vector Machine and Logistic Regression models*

TF-IDF is used to transform strings into a matrix of vectors, aiming to establish words importance assigning weights among them. Such action is decided by using statistics that take into account the number of times each word appears (Frequency) within the text corpora. TF-IDF features of numeric vectors will be used to fed SVM and LR classifiers.

$$\operatorname{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Figure 8: TF-IDF

5.2 *BOW for Naive Bayes and Random Forest models*

In short, BOW is the most basic technique for representing sequences of documents. Because features are extracted by counting the number of words appearances in those documents. Dissimilar to GloVe and TF-IDF models, BOW does not really bargain for words organization nor grammar. And it is said to be a good technique when used over small data. BOW features will be use alongside with NB and RF models.

$$BoW3 = BoW1 \biguplus BoW2$$

Figure 9: Bag Of Words

5.3 *Pre-trained GloVe Embeddings for Bi-LSTM model*

GloVe is a public unsupervised log-bilinear regression model that have been trained in advanced over large text corpora, using a word-word co-occurrence matrix of 300 dimensions. With the purpose of boosting future training times of similar tasks, by deriving the relationship between them words.

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

Figure 10: GloVe

In particular, GloVe will be used alongside with our Bi-LSTM model. Therefore, a *vocabulary* containing a list of all words in the movie reviews dataset, was created. Next, we check whether these words included in our *vocabulary* are actually recognised by GloVe. At first, and before running (pre-processing) tasks, we found GloVe embeddings only for 12% of the *vocabulary* and 90% of all reviews. Addressing this issue, we used GloVe pre-trained embedding itself and created a filter based on: a list of commonly characters (white list), another list containing known special characters recognized by GloVe and lastly, another one made of all special characters that are not recognized by GloVe. As a consequence, we could filter some noisy characters before starting the pre-processing phase. Additionally, during the pre-processing stage we needed to iterate back and forward, so that we could run additional pre-processing tasks in order to achieve higher coverage values. Such iteration increased our coverage to 57% of the *vocabulary* and 99% of all reviews. Now, the GloVe embedding layer is ready to be fed to the Bi-LSTM model, as it does now cover most portion of the reviews words (*Figure 11*).Data was split on a proportion of 70:30 for training and testing respectively, using *train_test_split* method from *sklearn* package.

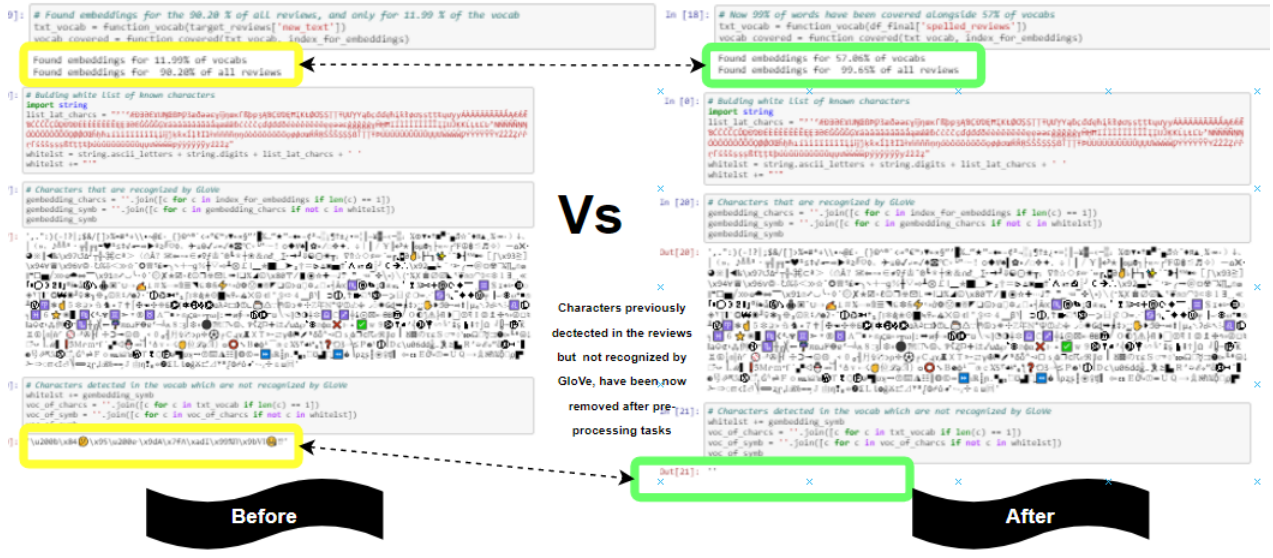


Figure 11: Filtering noisy characters using GloVe pre-trained embeddings

On the other hand, LSTM or Unidirectional LSTM are designed with short term memory as a means to address learning problems with sequential data. Nonetheless, it is just able to retain the information that has already passed through. Subsequently, Bi-LSTM architecture boosts LSTM's concept since allows the sequential learning signal propagation in two ways, ***forward*** remembering the past, and ***backward*** remembering the future. In simple terms, Bi-LSTM architecture is the result of assembling two LSTM's together. Furthermore, it is pertinent to clarify that one can't really assume that either architecture will improve the accuracy of a given model or not. Because up to the present time, there isn't any study proving that Bi-LSTM outperforms LSTM in 100% of applied cases and vice versa.

Our Bi-LSTM model runs Keras on top of Tensorflow, with a total of two Bidirectional layers with 128 units for deep learning purposes. This architecture uses a matrix of pre-trained GloVe

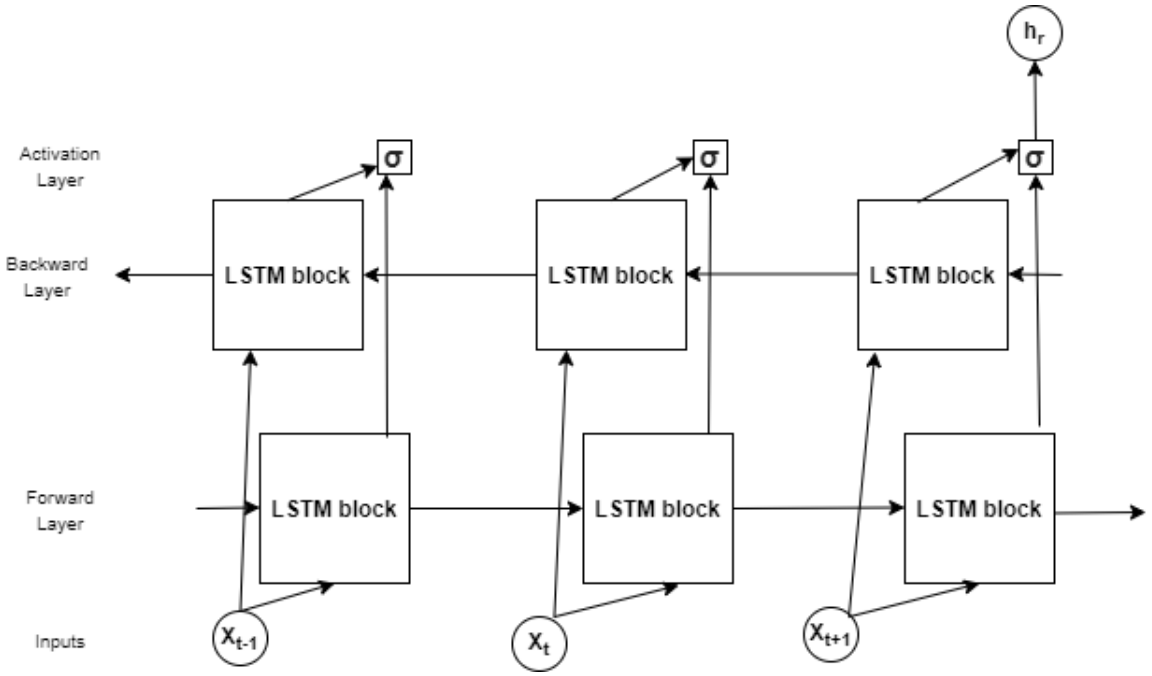


Figure 12: Bi-LSTM architecture

embeddings layer of 800B 300D dimensions (840B tokens, 2.2M vocab, cased, 300d vectors) weights for Keras. Trained with a batch of 100, using Adam optimizer and a Dropout layer to address over fitting issues. Altogether concatenates a hybrid Average Pooling and Max Pooling is implemented, with the aim of dimension reduction and computational complexity. Lastly, a Keras dense layer with Sigmoid activation is placed in the deep learning network architecture, in order to produce a transformation of the network node input signal into an output.

6 Evaluation

Be on our case, we've conducted 5 cases of study comprising 3 different word embedding techniques: SVM classifier with TF-IDF, LR classifier with TF-IDF, NB classifier with BOW, RF classifier with BOW and Bi-LSTM model using pre-trained GloVe word embeddings. In behalf of answering the novel proposed question, and assuming that dealing with online commentary for movie reviews is at issue. Our models have been evaluated based on two different aspects: *Area Under the Curve (AUC)* score and models *Wall Time* (training time).
Let us remark that Google Colab Cloud service has a restriction of 12 hrs per session, so it is important to note that following results were obtained from training and testing sets that only account for the 10% out of total processed data. So that the classification model demo applied over our large corpora of text, could run during these time frame without interruption.

6.1 Support Vector Machine with TF-IDF

Considering the fact that our data contains binary classes, our first experiment focused on building a SVM classifier which was trained using a Linear kernel, as it is well-known for delivering good results over documents, because of text including features that are considered to be linearly separable. Jointly, after completion of hyper-parameters tuning using grid search, we've included a regularization strength by 1.0 and gamma parameter set to *auto*.

6.2 Logistic Regression with TF-IDF

In Natural Language Processing, discriminative classifiers become pivotal for text analysis because supervised algorithms such as the LR allow to train the classifier in such way, so that can make adequate distinctions about the classes. In this sense, our LR classifier was optimized using Grid Search method, and trained resorting to L2 regularization (Ride Regression) in order to stabilize the estimates as it is assumed the presence of Collinearity along the data. Additionally a small C value was placed as reinforcement for stronger regularization.

6.3 Naive Bayes with BOW

In the third experiment we've chosen to train a Bernoulli Naive Bayes classifier. With the aim of improving the Recall value per class while keeping a fast computational processing ability to be at hand, using small amount of data as for the training stage. Bernoulli NB was specifically designated for this experiment due to the distinctly penalisation of non-occurrences rule, that this type of NB algorithm applies over binary features.

6.4 Random Forest with BOW

In this case, we attempted to built a good classifier for our text corpora, by appealing the potential of decision trees. On this basis we built a RF classifier to test the prediction capability of more than one decision tree (a forest of decision trees). So to fit the data, we've chosen to limit the total of trees by 100, with a maximum depth of each tree by 2.

6.5 Bi-LSTM with pre-trained GloVe

Lastly, for our proposed Bi-LSTM model we put in place a pre-trained GloVe word embedding layer with a co-occurrence matrix. Built from weighing the differences between the millions of words that already existed in the (pre-trained) GloVe model, versus the words present in our movie reviews pre-processed data. Next off, we fed our model with this originated co-occurrence matrix instead our previous text corpora data. Afterwards, we ran a deep learning model with two bidirectional layers of 128 units, using a fixed number of sequences that were trained together (BATCH_SIZE = 100), and finally passing the training data both forward and backward, through the LSTM network twice (epochs = 2).

6.6 Discussion

The *AUC* score is commonly used to assess binary classifications due to the fact that is able to distinguish between classes; by representing the probability that a model has successfully positioning random examples to the right side, either positive or negative. In simpler words AUC is able to place observation onto the right side, where the closer to 1 the more True Positive observations were correctly classified. Meaning less False Positive observations remained to be miss corrected and classified as False Positive. Whereas *Wall Time*, measures the total amount of time that our model code takes from the moment it was submitted to the CPU, to the moment when the process has finished executing. *Wall Time* is an important factor if we are visualising our Machine Learning model as a foundation work for future implementations, where processing speed for online movie commentary data, must be carried out quick.

		Precision	Recall	F1-score	Accuracy	AUC score	Wall Time
SVM - TFIDF	*Negative*	0.78	0.94	0.85	0.76	***0.60***	***40min 5sec***
	Positive	0.64	0.28	0.39			
LR - TFIDF	*Negative*	0.73	1.00	0.85	0.73	***0.5***	***644 msec***
	Positive	0.00	0.00	0.00			
NB - BOW	*Negative*	0.80	0.78	0.79	0.70	***0.62***	***57.7 msec***
	Positive	0.44	0.46	0.45			
RF - BOW	*Negative*	0.73	1.00	0.85	0.73	***0.5***	***839 msec***
	Positive	0.00	0.00	0.00			
Bi-LSTM - GloVe	*Negative*	0.79	0.93	0.86	0.77	***0.76***	***3min 48sec***
	Positive	0.64	0.33	0.44			

Table 3: Text classification models results

Viewed in this way, the SVM classifier actually achieved a fair *Precision* score by correctly classifying the class *'0' (Without spoiler)* with a score of 0.78. However, the classifier didn't perform as good over the opposite class *'1' (With spoiler)*, with only scoring a *Precision* of 0.64 (*Table 3*). Moreover, the AUC score of 0.60 indicates that this classifier has low separation capacity, proving a rather not good performance (Table 3). Additionally, the process of fitting the SVM classifier to the TF-IDF vectors, took 40 min. and 5 seg. of Wall Time for only 10% out of total data. This is considered another downside for this classifier, because we cannot lose sight of the fact that having a low measure of separability plus long training times over small portions of data, won't deliver a suitable solution for future approaches.

Likewise, perhaps attaining good *Precision* score for the class 0 (Without spoilers), our LR classifier performed really poor over class 1 (With Spoilers). Thus, showing no skills whatsoever in terms of predictions with an AUC score of 0.50. These lack of prediction ability is evidenced by the diagonal line in the AUC graph, which indicates that our predictions will be either as good as random. On the other hand, LR classifier shows a really good Wall Time value of 644 millisecond's (*Table 3*), which is a really good attribute for these kind of NLP approach.

Moreover, NB classifier obtained an AUC score of 0.62. This classifier also shows low skill to correctly classify over the class 1 (With spoilers). Oppositely to SVM's, NB classifier accomplished a Wall Time of only 57.7 millisecond's and become the quickest during the fitting stage out of all five (*Table 3*).

By the same token as the LR classifier, and despite the fact that RF algorithm is being categorised as robust and versatile with a proven effectiveness on predictive modeling. Our RF classifier achieved a poor AUC score of 0.50, hence evidencing lack of class separation ability (*Table 13*). Perhaps, the reason behind this poor performance lays on the fact that having used BOW technique to represent words, could've affected the performance of this classifier. Because RF algorithm is well-known for being sensible against high dimensional sparse data, and BOW method falls in this category. Even though, this model achieved a really good Wall time of 839 millisecond's, it's lack of ability of correctly predicting classes labels it as a bad classifier.

On the other side of the coin, our proposed model for classification of movie reviews, achieved the highest AUC score out of all five models. This is implied by the curvy shape line of the AUC graph (*Figure 13*) itself, which shows that our Bi-LSTM model using pre-trained GloVe technique, can predict high numbers of True Positives and a fairly number of True Negatives. Although, we can also deduce that the prediction of False Positives remains high, thus affecting the performance of our model. Evidently, this unveils a drawback for our model and with-it, room for improvement. As the case may be and given the circumstances that we are dealing with user-generated data, the limitation exposed by our model indwells in the pre-processing stage. Where a deeper grammar correction and

more strict named entity recognition strategy, can still be applied in order to increase the coverage of GloVe vocabulary. Conjointly, the Wall Time of 3 min and 48 sec delivered by our model, ratifies that despite the fact that NLP tasks and Deep Learning architecture altogether, demand a considerable computational power; the inclusion of pre-trained words shortens the learning stage of the model, by taking advantage of the pre-trained GloVe feature. Thus, demonstrating that our model has also succeed in delivering a good performance overall.

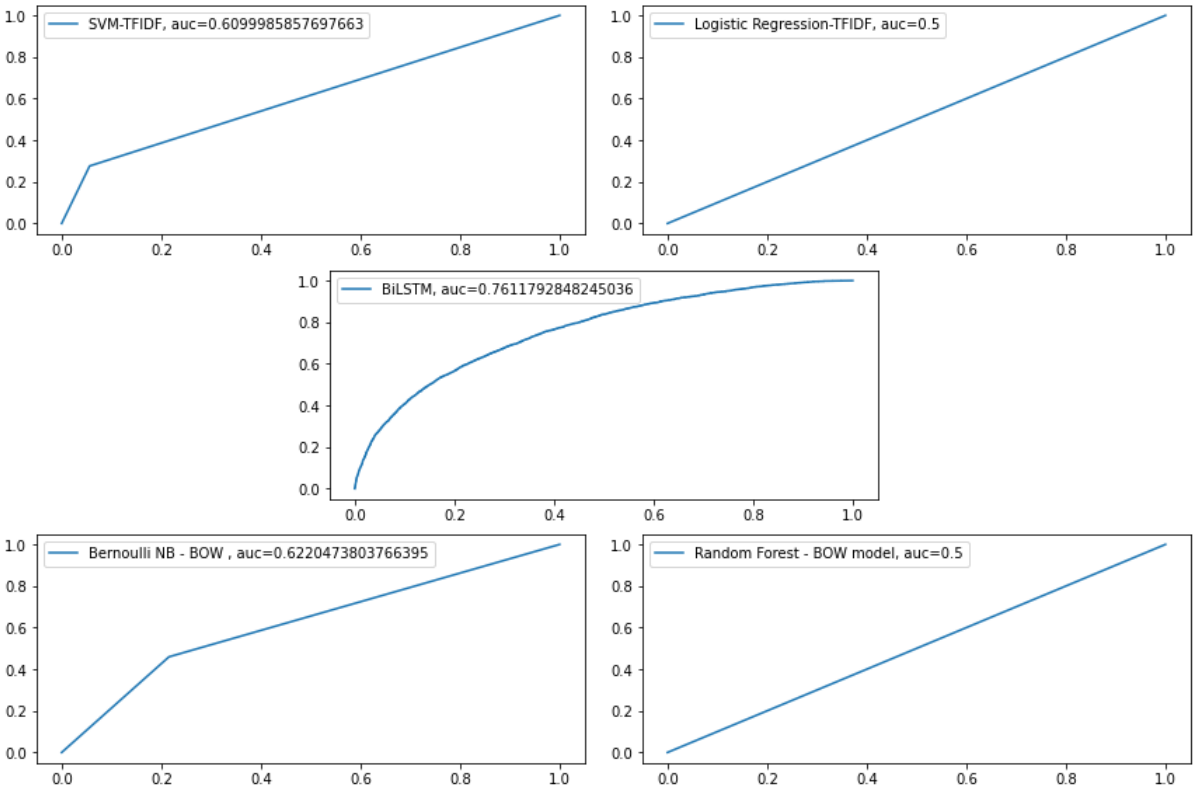


Figure 13: AUC scores for implemented models

7 Conclusion and Future Work

In this research, we proposed to build a Machine Learning model based on a type of Neural Network architecture. With the aim of serving as foundation work to address the modern issue that movie spoilers represent, for human well-being and film industry revenue themselves. The above-mentioned proposal, was successfully built combining the power of a deep learning architecture like LSTM, altogether with pre-trained GloVe model. The results obtained from our approach are considered to be satisfactory, since our model proved that the usage of Neural Networks and pre-trained word embeddings, can equally boost up the discrimination capacity between classes, while keeping low training times for text classifiers; that are needed to be designed for detection of spoilers throughout social sites for movies related fan. In total, we built and tested five different classifiers, our proposal included. Where we've evaluated how every model performed according to their AUC score and the total time that took to each classifier fitting the data. In this wise, we could demonstrate divergent levels of skill to distinguish between binary classes (True(1) or False (0)) that were delivered among the experiments. All things considered, we come to terms that our proposed model outperformed all the others evaluated models because not only succeed in classifying a big number of True Positives, but also completed the task in a suitable time lapse.

However, there are still tasks that can be done in order to improve our model. Just like future work, that could focus on taking the pre-processing of text stage to the next level. For the purpose of getting rid of all these extra noise that comes in the form of: grammar mistakes and name entities (movie names, brands, characters and actor names, etc); that we are still failing to remove even with the already executed tasks. For instance, including external services (e.g. Microsoft Language Understanding Intelligent Service L.U.I.S) that are expressly built to deal with NLP tasks, could help on increasing the coverage of the vocabulary for the co-occurrence matrix generated by the pre-trained GloVe method. Additionally, taking different languages into consideration during the pre-processing stage, could also increase the model performance.

References

Amit Mandelbaum, A. S. (2016). Word Embeddings and Their Use In Sentence Classification Tasks.

Basaldella, M., Antolli, E., Serra, G. and Tasso, C. (2018). Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction, (December): 0–8.

Catal, C. and Nangir, M. (2017). A sentiment classification model based on multiple classifiers, *Appl. Soft Comput. J.* **50**: 135–141.

Comparison of Naïve Bayes , Random Forest , Decision Tree , Support Vector Machines , and Logistic Regression Classifiers for Text Reviews Classification (2017). **5**(2): 221–232.

Ertugrul, A. M. (2018). Movie Genre Classification from Plot Summaries using Bidirectional LSTM, (February).

Escalante, H. J. (2014). A comparison of outlier detection algorithms for machine learning A Comparison of Outlier Detection Algorithms for Machine Learning.

Eyheramendy, S. (2019). On the Naive Bayes Model for Text Categorization, (December 2002).

Fersini, E., Messina, E. and Pozzi, F. A. (2016). Expressive signals in social media languages to improve polarity detection, *Inf. Process. Manag.* **52**(1): 20–35.

Guo, S. and Ramakrishnan, N. (2010). Finding the Storyteller : Automatic Spoiler Tagging using Linguistic Cues, *Comput. Lingyistics* (August): 412–420.

Ifrim, G. and Weikum, G. (2008). Fast Logistic Regression for Text Categorization with Categories and Subject Descriptors, (August).

Jin, P., Zhang, Y., Chen, X. and Xia, Y. (2013). Bag-of-Embeddings for Text Classification, pp. 2824–2830.

Johnson, B. K. and Rosenbaum, J. E. (2015). Spoiler alert: Consequences of narrative spoilers for dimensions of enjoyment, appreciation, and transportation, *Communication Research* **42**(8).

Kannan, R., Woo, H. and Park, H. (2017). Outlier Detection for Text Data : An Extended Version.

Learning, M.-t., Liu, P., Qiu, X. and Huang, X. (2011). Recurrent Neural Network for Text Classification.

Leavitt, J. D. and Christenfeld, N. J. S. (2014). The fluency of spoilers, *John Benjamins Publishing Company* (September).

Liu, Y., Liu, Z., Chua, T.-s. and Sun, M. (2015). Topical Word Embeddings, pp. 2418–2424.

Luo, F. (2016). Affective-feature-based Sentiment Analysis using SVM Classifier, *2016 IEEE 20th Int. Conf. Comput. Support. Coop. Work Des.* pp. 276–281.

Ma, C., Xu, W., Li, P. and Yan, Y. (2015). Distributional Representations of Words for Short Text Classification, (21): 33–38.

Mahdi, S., Rahmani, R., Ghodsi, A. and Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings, *Expert Syst. Appl.* **117**: 139–147.

Nakashole, N. (2019). Fine-Grained Spoiler Detection from Large-Scale Review Corpora.

Pennington, J., Socher, R. and Manning, C. (2014). GloVe : Global Vectors for Word Representation.

Rezaeinia, S., Ghodsi, A. and Rahmani, R. (2017). Improving the accuracy of pre-trained word embeddings for sentiment analysis.

Xu, B. (2012). An Improved Random Forest Classifier for Text Categorization, **7**(12): 2913–2920.

Xu, K., Zhou, Z., Hao, T. and Liu, W. (2018). A Bidirectional LSTM and Conditional Random Fields Approach to Medical Named Entity Recognition A Bidirectional LSTM and Conditional Random Fields, (December 2017).

Zhang, D., Xu, H., Su, Z. and Xu, Y. (2015). Expert Systems with Applications Chinese comments sentiment classification based on word2vec, *Expert Syst. Appl.* **42**(4): 1857–1863.

Zhou, P. and Qi, Z. (2016). Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling, **2**(1): 3485–3495.