Sentiment Analysis Using Deep Learning
(Review 1)

Sima Shafaei & Adriana Ontiveros Gonzalez

Feb 17

1 Introduction

Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, emotions, appraisals, and attitudes towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes (Liu, 2015)

Since early 2000, as the Web and social media have grown, sentiment analysis has become one of the most active areas of research in AI and Natural Language processing. Nowadays, we can collect a huge volume of data from reviews, forum discussions, blogs, microblogs, Twitter, and social networks that is useful not only for individuals but also for organizations. In today's world, if one wants to buy a new product, there is more opportunity than only asking about it from friends and family because there is a lot of useful information on everything we want such as users reviews and discussions on the internet that makes us aware of its strengths and weaknesses. Organizations can also use this information to reshape their business, improve their products, and better meet customer demands.

Studies in sentiment analysis can be divided into three general categories: document level, sentence level, and aspect level. Document-level sentiment classification categorizes an opinionated document (e.g., a product review) as expressing an overall positive or negative opinion. Sentence-level sentiment classification categorizes individual sentences in a document. However, each sentence cannot be assumed to be opinionated. Traditionally, one often first classifies a sentence as opinionated or not opinionated, which is called subjectivity classification. Then the resulting opinionated sentences are classified as expressing positive or negative opinions. For example, in a product review, aims to summarize positive and negative opinions about different aspects of the product respectively, although the general sentiment on the product could be positive or negative. (Zhang, Wang, & Liu, 2018)

The topic we are considering for this project falls into the document-level sentiment analysis category. The remainder of this paper is organized as follows. Section 2 summarizes related works and state of the art articles, their methods, and architectures. Section 3 introduces common features that feed the first layer of the neural network for the sentiment analysis problem. Section 4 provides information about several available datasets used in articles and their statistical information. Section 5 presents different metrics and measures used in previous works for evaluating document level sentiment analysis. Section 6 describes the current project case study, selected dataset, and evaluation metrics. Section 7 provides an intuitive sense of our dataset by creating graphics and statistical information to describe features, their properties and relationships. Finally, in section 8, we present an activity calendar that explains our future activity per week.

2 Previous Works

Research on sentiment analysis can be divided into two categories. The first group is the studies follow (Pang, Lee, & Vaithyanathan, 2002) and work on designing effective features for building a powerful sentiment classifier. Representative features include word n-grams(Wang & Manning, 2012), text topic (Ganu, Elhadad, & Marian, 2009), bag-of-opinions (Qu, Ifrim, & Weikum, 2010), syntactic relations (Xia & Zong, 2010), sentiment lexicon features (Kiritchenko, Zhu, & Mohammad, 2014). The second set of studies uses machine learning methods in a supervised learning framework and is concerned with improving different learning methods.

(Turney, 2001) introduces an unsupervised approach by using sentiment words/phrases extracted from syntactic patterns to determine the document polarity.(Goldberg & Zhu, 2006) place this task in a semi-supervised setting, and use un-labelled reviews with graph-based method. (Moraes, Valiati, & Gavião Neto, 2013) made an empirical comparison between SVM and ANN for document level sentiment classification, which demonstrated that ANN produced competitive results to SVMs in most cases

Since the subject of this project is document-level sentiment analysis using a deep learning method, in the following, we discuss more details and architecture of several state-of-the-art articles in this area. Most of these methods improved the architecture of LSTM or CNN for the sentiment analysis application. LSTM is supposed to capture the long-term and short-term dependencies simultaneously, but when dealing with considerably long texts, LSTM also fails on capturing and understanding significant sentiment message.

To solve this problem (Xu, Chen, Qiu, & Huang, 2016) proposes a cached long short-term memory neural networks (CLSTM) to capture information in a longer step by introducing a cache mechanism. Moreover, in order to better control and balance the historical message and the incoming information, they adopt one particular variant of LSTM proposed by (Greff, Srivastava, Koutnik, Steunebrink, & Schmidhuber, 2017) the Coupled Input and Forget Gate LSTM (CIFG-LSTM). Figure 1 gives an overview of the architecture of their model. Since the first group, the slowest group, is supposed to keep the long-term information and can better represent a whole document, they only utilize the final state of this group to represent a document. They concatenate the state of the first group in the forward LSTM at T^{th} time-step and the first group in the backward LSTM at the first time-step.

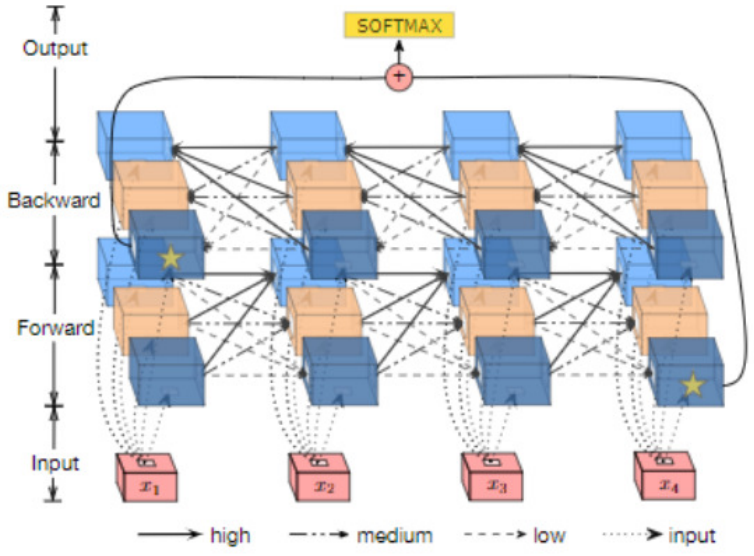


Figure 1: An overview of the proposed architecture by (Xu et al., 2016). Different styles of arrows indicate different forgetting rates. Groups with stars are fed to a fully connected layers for softmax classification. Here is an instance of B-CLSTM with text length equal to 4 and the number of memory groups is 3

Another limitation of the standard LSTM architectures is that they only allow for strictly sequential information propagation. In (Tai, Socher, & Manning, 2015) authors introduce a generalization of the standard LSTM architecture to tree-structured network topologies and show its superiority for representing sentence meaning over a sequential LSTM. This work proposes two natural extensions to the basic LSTM architecture: The Child-Sum Tree-LSTM and the N-ary Tree-LSTM. Both variants allow for richer network topologies where each LSTM unit is able to incorporate information from multiple child units.

While the standard LSTM composes its hidden state from the input at the current time step and the hidden state of the LSTM unit in the previous time step, the tree-structured LSTM, or Tree-LSTM, composes its state from an input vector and the hidden states of arbitrarily many child units. The standard LSTM can then be considered a special case of the Tree-LSTM where each internal node has exactly one child. They show that Tree-LSTMs outperform strong LSTM baselines on two tasks: predicting the semantic relatedness of two sentences and sentiment classification. Implementations of this models and experiments are available at https://github.com/stanfordnlp/treelstm

(Tang, Qin, & Liu, 2015) Address the challenge of encoding the intrinsic relations between sentences in the semantic meaning of a document. This approach models document semantics based on the principle of compositionality, which states that the meaning of a longer expression (e.g. a sentence or a document) comes from the meanings of its constituents and the rules used to combine them. Since a document consists of a list of sentences and each sentence is made up of a list of words, the approach models document representation in two stages.

Authors introduce a neural network model to learn vector-based document representation in a unified, bottom-up fashion. The model first learns sentence representation with convolutional neural network or long short-term memory. Afterwards, semantics of sentences and their relations are adaptively encoded in document representation with gated recurrent neural networks. Document representations are then used as features for document level sentiment classification. Their experimental results show that this neural model shows superior performances over several state-of-the-art algorithms and gated recurrent neural network outperforms standard RNN in document modeling for sentiment classification. An overview of this approach is displayed in Figure 2

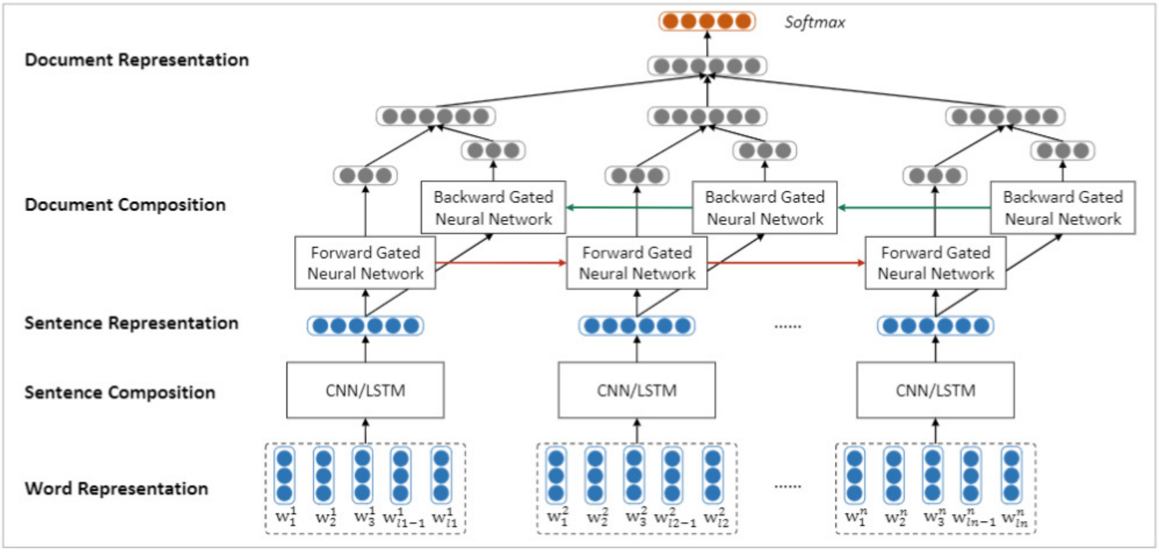

Softmax
Document Representation
Document Composition
Backward Gated Neural Network
Backward Gated Neural Network
Backward Gated Neural Network
Forward Gated Neural Network
Forward Gated Neural Network
Forward Gated Neural Network
Sentence Representation
......
Sentence Composition
CNN/LSTM
CNN/LSTM
CNN/LSTM
Word Representation
......
w_1^1 w_2^1 w_3^1 w_{l1-1}^1 w_{l1}^1
w_1^2 w_2^2 w_3^2 w_{l2-1}^2 w_{l2}^2
w_1^n w_2^n w_3^n w_{ln-1}^n w_{ln}^n

Figure 2: The neural network model for document level sentiment classification. w_i^n stands for the i-th word in the n-th sentence, l_n is sentence length (Tang et al., 2015)

In (Zhai & Zhang, 2016), authors investigate the usage of auto-encoders in modeling textual data. To address problems of traditional auto-encoders which suffers from scalability with the high dimensionality of vocabulary size and dealing with task-irrelevant words, this paper learns a task-specific representation of the textual data by relaxing the loss function in the auto-encoder to the Bregman divergence and also derives a discriminative loss function from the label information.

In particular, they first train a linear classifier on the labeled data, then define a loss for the auto-encoder with the weights learned from the linear classifier. To reduce the bias brought by one single classifier, authors define a posterior probability distribution on the weights of the classifier, and derive the marginalized loss of the auto-encoder with Laplace approximation. This model is able to take advantage of unlabeled dataset and get improved performance.

3 Features

In this section, we describe the features that have been extracted from text in most sentiment analysis articles and used as the input of neural networks or other classical learning methods.

3.1 BoW (Bag of Words)

In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order. which means that two documents can have exactly the same representation as long as they share the same words.

In this representation, a document is transformed to a numeric feature vector with a fixed length, each element of which can be the word occurrence (absence or presence), word frequency, or TF– IDF score. Its dimension equals the size of the vocabulary. A document vector from BoW is normally very sparse since a single document only contains a small number of words in a vocabulary. Despite its popularity, BoW has some disadvantages. In addition to ignoring word order, BoW can barely encode the semantics of words. For example, the words “smart,” “clever,” and “book” are of equal distance between them in BoW, but “smart” should be closer to “clever” than “book” semantically.

3.2 Bag-of-n-grams

Bag of n grams is an extension for BoW. It can consider the word order in a short context (n-gram), but it also suffers from data sparsity and high dimensionality.

3.3 Word embedding

Word embedding is another popular representation of document vocabulary which is capable of capturing context of a word in a document, semantic and syntactic similarity and relation with other words. This method maps words into a list of real numbers and represents them in a dense (or low-dimension) vectors. For example, for the word "Hamburger" it creates a list of 64 numbers which these numbers describe this word in a new 64 dimensional space. These values carry some meaning in a way that the vector obtained for the word "cheeseburger" represents a point in this 64 dimensional space very close to the vector of "Hamburger".

Methods to generate this mapping include neural networks, dimensionality reduction on the word co-occurrence matrix, probabilistic models, explainable knowledge base method, and explicit representation in terms of the context in which words appear(Levy 2014)

4 Available datasets for sentiment analysis

In this section, we present some of the datasets used in previous work

4.1 Stanford Sentiment Treebank (Socher et al., 2013)

The corpus of movie review excerpts from the rottentomatoes.com website originally collected and published by (Pang & Lee, 2008). The original dataset includes 10,662 sentences, half of which were considered positive and the other half negative. Each label is extracted from a longer movie review and reflects the writer's overall intention for this review. The normalized, lower-cased text is first used to recover, from the original website, the text with capitalization. Remaining HTML tags and sentences that are not in English are deleted. The Stanford Parser (Klein & Manning, 2003) is used to parse all 10,662 sentences.

4.2 Yelp 2013, Yelp 2014 and Yelp 2015

review datasets derived from respectively Yelp Dataset Challenge of year 2013, 2014 and 2015. The sentiment polarity of each review is 1 star to 5 stars, which reveals the consumers' attitude and opinion towards the restaurants. These datasets are available at http://www.yelp.com/dataset_challenge. Statistical information of these datasets are shown in Table 1.

Table 1: Statistical information of Yelp 2013/2014/2015 datasets #docs is the number of documents, #s/d and #w/d represent average number of sentence and words contained per document, |V| is the vocabulary size of words, #class is the number of classes

Corpus	#docs	#s/d	#w/d	|V|	#class	Class Distribution
Yelp 2013	335018	8.90	151.6	211245	5	.09/.09/.14/.33/.36
Yelp 2014	1125457	9.22	156.9	476191	5	.10/.09/.15/.30/.36
Yelp 2015	1569264	8.97	151.9	612636	5	.10/.09/.14/.30/.37

4.3 IMDB

Popular movie review dataset consists of 348415 movie reviews ranging from 1 to 10. Average length of each review is 325.6 words, which is much larger than many review datasets a sample of IMDB dataset is available at http://ir.hit.edu.cn~/dytang/paper/acl2015/dataset.7z. Statistical information of this dataset is shown in Table 2

Table 2: Statistical information of IMDB datasets #docs is the number of documents, #s/d and #w/d represent average number of sentence and words contained per document, |V| is the vocabulary size of words, #class is the number of classes

Corpus	#docs	#s/d	#w/d	|V|	#class	Class Distribution
IMDB	348415	14.02	325.6	115831	10	.07/.04/.05/.05/.08/.11/.15/.17/.12/.18

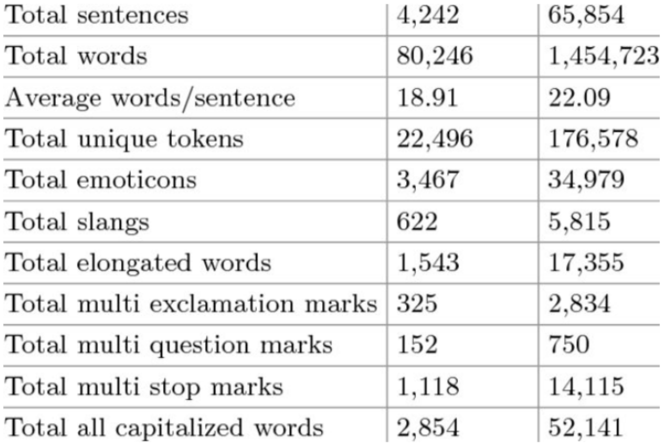
4.4 Amazon review

Amazon product reviews for different product types such as books, DVDs, electronics, kitchen appliances, and etc. Each review consists of a rating (0-5 stars), a reviewer name and location, a product name, a review title and date, and the review text. Reviews with rating > 3 are labeled positive, those with rating < 3 are labeled negative, and the rest discarded because their polarity was ambiguous.

4.5 Semeval-2015

SemEval (Semantic Evaluation) is an ongoing series of evaluations of computational semantic analysis systems, organized under the umbrella of SIGLEX, the Special Interest Group on the Lexicon of the Association for Computational Linguistics. SemEval has evolved from the SensEval word sense disambiguation evaluation series. This dataset contains sub tasks and collect data from different source (eg. Amazon, Twitter). Table 3 shows some statistical information about SS-Twitter SemEval

Table 3: Statistical information about SS-Twitter SemEval



Total sentences	4,242	65,854
Total words	80,246	1,454,723
Average words/sentence	18.91	22.09
Total unique tokens	22,496	176,578
Total emoticons	3,467	34,979
Total slangs	622	5,815
Total elongated words	1,543	17,355
Total multi exclamation marks	325	2,834
Total multi question marks	152	750
Total multi stop marks	1,118	14,115
Total all capitalized words	2,854	52,141

5 Metrics and measures

In all of the articles we reviewed, two following metrics were used to evaluate the proposed method

Accuracy: Accuracy is a standard metric to measure the overall classification result

$$Accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} = \frac{TP + TN}{TP + FP + TN + FN}$$

Mean Squared Error (MSE) is used to figure out the divergences between predicted sentiment labels and the ground truth ones

$$MSE = \frac{\sum_i^N (gold_i - predected_i)^2}{N}$$

6 Case Study

Sentiment classification at the document level is to assign an overall sentiment orientation/polarity to an opinion document, that is, to determine whether the document (e.g., a full online review) conveys an overall positive or negative opinion. In this setting, it is a binary classification task. It can also be formulated as a regression task, for example, to infer an overall rating score from 1 to 5 stars for the review. Some researchers also treat this as a five-class classification task. (Zhang et al., 2018)

In this project, we defined sentiment analysis as a binary classification task to determine the polarity on different reviews. From Amazon reviews in the range of May 1996 to October 2018 provided by the University of California San Diego (UCSD), the product category of Kindle Store was selected to work on, and just a small subset of the original data was obtained to be this the dataset of the project.

In regards to the reviews, it was taken into consideration only ratings 1 and 5, with the assumption that in a review of 5 which is the best, there should not be a negative review, and that in a review of 1 which is the worst, there should not be a positive review. The ratings 1 were mapped as 0, and the ratings 5 were mapped as 1.

7 Description of dataset

The dataset selected for this project is part of the Amazon Review Data (2018) provided by the University of California San Diego (UCSD), which includes 233.1 million of Amazon's client's reviews divided by product categories, these reviews are in the range May 1996 to October 2018 (Ni, Li, & McAuley, 2019).

For the purpose of this project, a "small" subset of the original data will be used for experimentation, and the product category selected was "Kindle Store," with 2,222,983 reviews. The format of the data comes in one-review-per-line in JSON format, and to parse it, a python code given by UCSD was used (Ni, Li, & McAuley, 2019).

The features included in the sample review are listed in Table 4.

Table 4: Features and their description for the Kindle Store dataset (Ni, Li, & McAuley, 2019)

Feature	**Description**
reviewerID	ID of the reviewer.
asin	ID of the product.
reviewerName	Name of the reviewer.
vote	Helpful votes of the review.
style	A dictionary of the product metadata.
reviewText	Text of the review.
overall	Rating of the product.
summary	Summary of the review.
unixReviewTime	Time of the review in Unix time.
reviewTime	Time of the review (raw).

In this case, as we will be performing a level sentiment analysis, only the features “reviewText” and “overall” were kept. The ratings in the overall feature go from 1 which is the worst rating to 5, which is the best rating. The number of reviews per category is listed in Table 5

Table 5: Number of reviews per rating

Rating	**Number of Reviews**
5	1,353,641
4	556,324
3	197,949
2	66,898
1	48,171

For the purpose of defining which is a positive review and which is a negative one, only the ratings 1 for negative reviews and 5 for positive reviews were kept, having the supposition that in a review of 5 which is the best, there should not be a negative review, and that in a review of 1 which is the worst, there should not be a positive review. Also, the remaining ratings were mapped in classes, 1 was given for ratings 5 and 0 was given for ratings 1.

Then, the dataset was reviewed for empty values, the overall feature was all filled, but the "reviewText" feature contained 297 empty values, so all those samples were deleted. After this, 1,353,349 reviews remained for class 1 and 48,166 remained for class 0.

The dataset was too imbalanced to work on it, so random down-sampling was applied on the dataset, and the majority class (1) was reduced to the same number of samples contained in the minority class (0). Finally, this is the dataset we will be working on.

A language detector was applied on the text of the reviews to define their language, although most of the reviews were in English, 32 different languages were detected in total (including Spanish, French, Italian, German, Swedish, among others). The majority of the languages were English with 93,189 reviews, compared to the sum of all the reviews in a language other than English, which was 3,143 reviews. This difference in languages can be visualized on the pie chart in Figure 3.

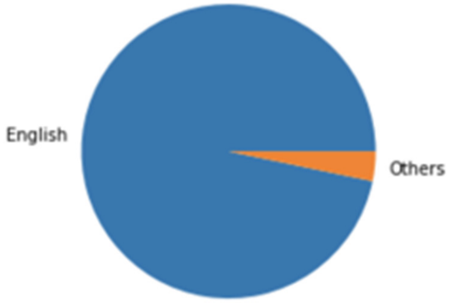


Figure 3: Pie chart of English Language vs Others in reviews for Kindle Store

With the purpose of maintaining only one language, all the samples with reviews written in a language different than English were deleted, leaving 46,817 for class 1 and 46,372 for class 0.

The data was then randomly split into a training and testing dataset, 20% for the testing and 80% for the training. It was also stratified based on the classes, which is the "overall" feature.

Afterward, some cleaning on the text of the reviews (on the training and testing datasets separately) was applied to be able to create a bag of words. The text was Unicode, then all was converted to lowercase, the punctuation and digits were removed, also the stemming was applied to all the words, and the stop-words were removed. The average number of words contained in the reviews was 46 words.

A dictionary of words for the training dataset was then created for each of the classes, containing the count or frequency for every word. Each of the dictionaries contains over 1.5 million words.

Figure 4, shows a horizontal histogram with the count of the 20 most frequent words from the positive reviews.

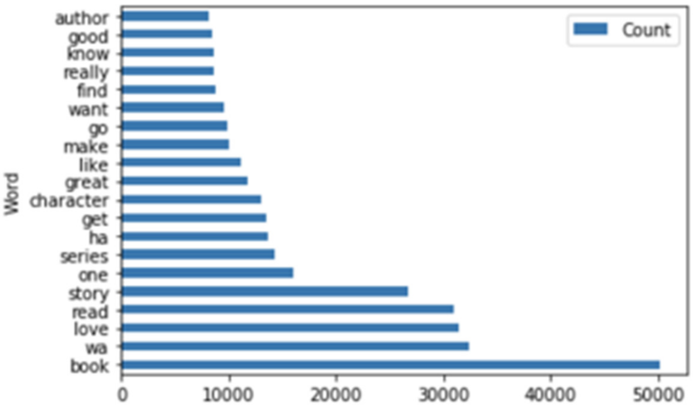


Figure 4: Horizontal histogram with 20 most frequent words from positive reviews

Figure 5, shows a horizontal histogram with the count of the 20 most frequent words from the negative reviews.

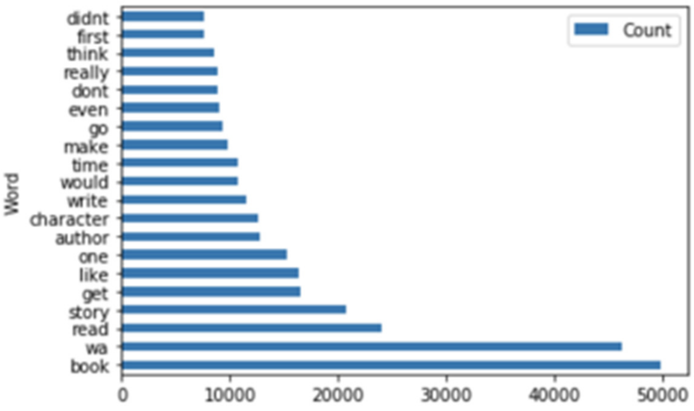


Figure 5: Horizontal histogram with 20 most frequent words from negative reviews

It is interesting to see, how the most common word for both of the reviews is book, which makes sense as Kindle Store is about books, this might indicate that it could be a good idea to eliminate this word from the dictionary, as the frequency in both of them is the same and this could be noise for the model.

Some other words are also in both of the sets, but their frequencies start to change. It can also be noticed that one of the most frequent words from the positive set is love, and on the other side, the negative set starts to contain words such as “didnt.”

More cleaning of the dataset might be needed to be able to get prediction results with a good accuracy.

8 Activity Calendar

Figure *6* shows the Gantt chart of the activities planned for the project. The index of colors on the right side of the chart indicates the percentage of completion of the tasks, so those tasks 100% completed, are shown in lighter green, and those with 0% completion are shown in darker green.

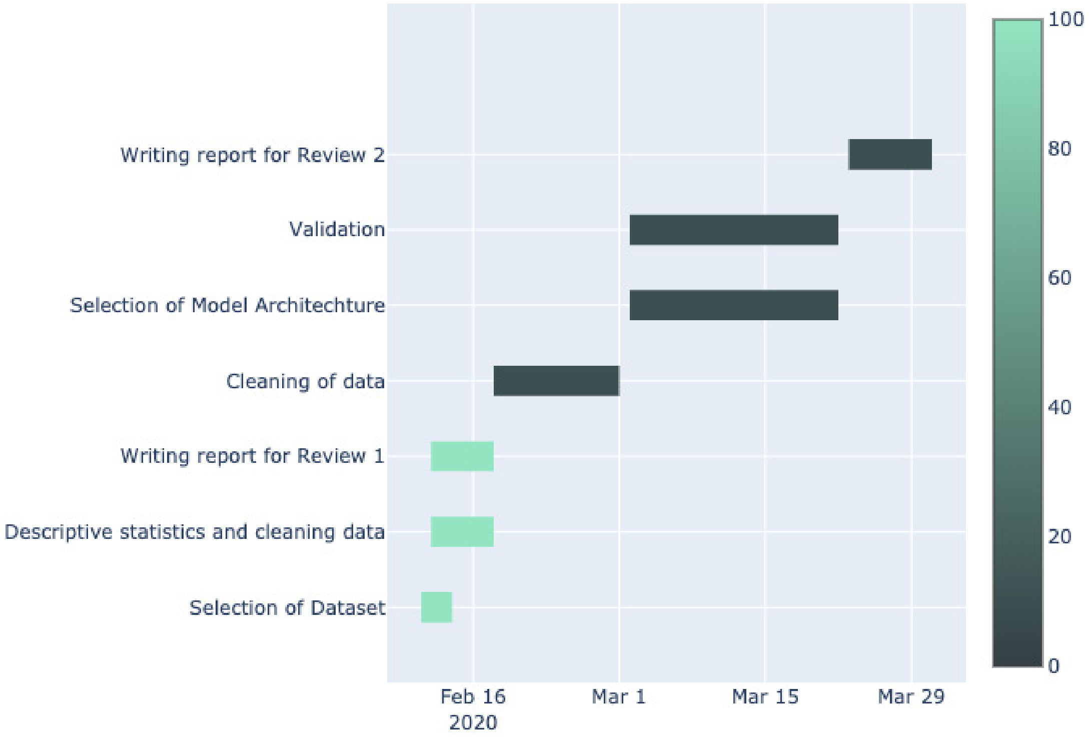


Figure 6: *Gantt chart of activities for the Final Project.*

9 References

Ganu, G., Elhadad, N., & Marian, A. (2009). Beyond the Stars : Improving Rating Predictions using Review Text Content. *WebDB*.

Goldberg, A., & Zhu, X. (2006). Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. *Proceedings of TextGraphs: The First*

Workshop on Graph Based Methods for Natural Language Processing. https://doi.org/10.3115/1654758.1654769

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2016.2582924

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*. https://doi.org/10.1613/jair.4272

Klein, D., & Manning, C. D. (2003). *Accurate unlexicalized parsing*. https://doi.org/10.3115/1075096.1075150

Liu, B. (2015). Sentiment analysis: Mining opinions, sentiments, and emotions. In *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. https://doi.org/10.1017/CBO9781139084789

Moraes, R., Valiati, J. F., & Gavião Neto, W. P. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2012.07.059

Ni, J., Li, J., & McAuley, J. (2019). *Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects*. https://doi.org/10.18653/v1/d19-1018

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*. https://doi.org/10.1561/1500000001

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques Bo. *Antike Und Abendland*.

Qu, L., Ifrim, G., & Weikum, G. (2010). The bag-of-opinions method for review rating prediction from sparse text patterns. *Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference*.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*.

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-Term memory networks. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.3115/v1/p15-1150

Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/d15-1167

Turney, P. D. (2001). *Thumbs up or thumbs down?* https://doi.org/10.3115/1073083.1073153

Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. *50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 - Proceedings of the Conference*.

Xia, R., & Zong, C. (2010). Exploring the use of word relation features for sentiment classification. *Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference*.

Xu, J., Chen, D., Qiu, X., & Huang, X. (2016). Cached long short-term memory neural networks for document-level sentiment classification. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. https://doi.org/10.18653/v1/d16-1172

Zhai, S., & Zhang, Z. (2016). Semisupervised autoencoder for sentiment analysis. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. https://doi.org/10.1002/widm.1253