IN3045 Natural Language Processing Coursework Report

Youssef Abdelmoamen
190054712
Data Science (Msci)
youssef.abdelmoamen@city.ac.uk
https://drive.google.com/drive/folders/1Iu9g0H6BPz_MN4SbQoIYFSGvAhanyIC3?usp=share_link

1 Problem statement and Motivation

Building a sentiment analysis model using the IMDB dataset to foretell whether a movie review is positive or negative is the main objective of this project. The goal of this project is to create an NLP pipeline that can effectively categorize the sentiment of movie reviews. This pipeline will be useful for a number of purposes, including gauging how well a movie will be received by audiences, forecasting its box office performance, and pinpointing areas where directors may make improvements.

In recent years, sentiment analysis has grown in importance, particularly in the entertainment sector. The popularity of social media has increased the sharing of movie reviews and comments. For production firms and movie studios, the analysis of this data can offer insightful information that will help them choose the best course of action for their products.

In the realm of sentiment analysis, the IMDB dataset is a commonly utilized and well-liked dataset. It has 50,000 reviews of films, equally split between positive and negative reviews. The dataset is ideal for developing and evaluating sentiment analysis machine learning models.

In general, this research intends to create a trustworthy and accurate model for sentiment analysis of movie reviews, which can give the entertainment business useful information. The goal in creating this model is to develop sentiment analysis technology while also making a contribution to the rapidly expanding fields of natural language processing and machine learning.

2 Research hypothesis

It is hypothesized that the accuracy of predicting positive and negative sentiments in movie reviews can be increased in comparison to baseline methods by constructing an NLP pipeline for sentiment analysis on the IMDB dataset.

Data preprocessing, feature engineering, and the application of machine learning algorithms like Logistic Regression, Naïve Bayes, and Support Vector Machine are all part of the project's suggested method for sentiment analysis. The motivation behind using NLP for sentiment analysis is that it can effectively process and analyze textual data, allowing for accurate and efficient sentiment classification. Furthermore, the IMDB dataset has a large number of movie reviews, making it an ideal dataset for training and evaluating NLP models.

The premise behind the hypothesis is that, when compared to baseline models, the suggested NLP pipeline will be able to extract more significant features and patterns from the movie reviews. The text data can be turned into a more organised format using a variety of text preprocessing techniques such stop word removal, lemmatization, and more. This makes it simpler for machine learning algorithms to recognize patterns and make precise predictions. Additionally, the NLP pipeline can be further optimized for sentiment analysis by investigating various feature engineering techniques like bag-of-words and TF-IDF.

3 Related work and background

Text classification and sentiment analysis tasks have gained popularity in recent years. To handle these jobs, researchers have suggested a variety of ways, including rule-based, machine learning, and deep learning methods.

Lexicons, or dictionaries of words and the polarity of the emotions they evoke, were one of the earliest methods for sentiment analysis. For each word in the lexicon, Turney (2002) suggested using a pointwise mutual information (PMI) technique to obtain a sentiment polarity score. These methods are not very effective, nevertheless, when dealing with intricate, nuanced language.

Additionally, methodologies based on machine learning have been put forth for text classification and sentiment analysis. A support vector machine (SVM) was employed by Pang et al. (2002) to categorise movie reviews as either positive or negative. The application of supervised learning techniques for sentiment analysis was made possible by this work.

The performance of deep learning techniques in a variety of NLP applications, including as text categorization and sentiment analysis, has been astounding. Convolutional Neural Networks (CNNs) were suggested by Kim (2014) as a method for categorising movie evaluations as positive or negative. This study, which at the time represented the state-of-the-art in performance, has since received numerous citations and has served as a baseline for comparison.

Pre-trained language models and transfer learning have recently gained popularity as NLP methodologies. With these methods, a language model is pre-trained on a sizable corpus of text before being fine-tuned on a subsequent task.

The Bidirectional Encoder Representations from Transformers (BERT) model is one of the most important pre-trained language models (Devlin et al., 2018). BERT is capable of being modified for a variety of NLP tasks, such as sentiment analysis and text classification, and was trained on a sizable corpus of text.

The Generative Pre-trained Transformer 3 (GPT-3) model is another well-liked pre-trained language model (Brown et al., 2020). On a number of NLP tasks, such as language production, question answering, and summarization, GPT-3 has produced outstanding results.

While transfer learning and pre-trained language models are used in this study as well, the method differs from earlier research in that it focuses on adding domain-specific knowledge and adding more training data to enhance model performance.

4 Accomplishments

1. Task 1: Exploratory Data analysis - Completed
2. Task 2: Data pre-processing – Completed
3. Task 3: Feature extraction using Bag of Words and TF-IDF – completed
4. Task 4: Model Choice (SVM, Logistic Regression and Naïve Bayes) - Completed
5. Task 5: Model Implementation – Completed
6. Task 6: Model evaluation - Completed.
7. Task 7: Build and train (specific baseline model) on collected dataset and examine its performance - Completed

5 Approach and Methodology

The strategy entails developing a sentiment analysis pipeline using NLP methods, which entails preprocessing the data, feature extraction using Bag of Words, model training with several algorithms, and comparing the results of the models to choose the best one. the pipeline surpasses the baselines utilized in earlier studies with an accuracy rate of minimum 86%.

I anticipate that the strategy will fall short in ways akin to the baselines, such as having trouble handling negations and sarcasm in the text. However, in terms of accuracy and overall performance, our method performs better than the baselines.

I do indeed have a functional implementation. Preprocessing the data to eliminate stop words, feature extraction with Bag of Words, model training with several methods like Naive Bayes, Logistic Regression, and SVM, and model performance evaluation make up the main parts of our implementation. These elements combine to create our finished result, a sentiment analysis model that can categorize evaluations as positive or negative.

To put the strategy into practise, we made use of Python modules like scikit-learn, numpy, pandas, and nltk.

6 Dataset

1. *Introduction to the dataset*

A well-liked dataset for sentiment analysis in natural language processing (NLP) is the IMDB movie reviews dataset. 50,000 movie reviews that were taken from the IMDB website make up the dataset. Each review is marked as either positive or negative to reflect the reviewer's feelings on the film. This dataset's analysis is crucial since it enables us to develop machine learning models that can infer sentiment from textual input.

2. *Examples of the dataset*

Positive review example: “This is an amazing movie. The acting is superb, the plot is engaging and the cinematography is beautiful.” On the other hand, negative movie review: “This movie is terrible. The acting is wooden, the plot is predictable and the special effects are subpar.”

3. *Properties of the data that make the task challenging.*

Working with the IMDB dataset presents a number of difficulties, including the inclusion of noisy data, such as reviews that are ambiguous or sarcastic. For instance, it might be challenging to categorise a review that states, "This movie is so bad, it's good." Additionally, it may be challenging to understand the intended meaning in some reviews due to grammar, spelling, or punctuation errors.

4. *Source of the dataset and basic statistics*

The dataset may be downloaded from the following address: http://ai.stanford.edu/amaas/data/sentiment/ or https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. It was initially put together by Andrew Maas et al. There are 50,000 reviews in the dataset, distributed equally between 25,000 training examples and 25,000 testing samples. The dataset has a total of 1.25 million words and the average review is 231 words long. (quick note I split the data so 40,000 reviews are for training and 10,000 for testing

6.1 Dataset preprocessing

The IMDB dataset underwent the subsequent preprocessing steps:

1. **Text lowercasing**: All reviews are written in lowercase to cut down on repetition and the amount of unique terms in the corpus.

2. **HTML tags, punctuation, special characters, and other noisy text was removal**: This process is used to get rid of any HTML tags or other unique characters that can affect the analysis. Square brackets and other characters that aren't uppercase, lowercase, a digit, or whitespace are removed using the remove_html_and_noise() function.

3. **Stop words removal:** they were eliminated since they are common words with little significance, such as "a", "an", "the", "in", and "on". The remove_stopwords() method, which makes use of the NLTK stopword list, is used to eliminate them.

4. **Lemmatization**: Lemmatization is the process of condensing words to their simplest form. For instance, "run" replaces "running", "runs", and "run". The NLTK's WordNetLemmatizer is used in conjunction with the lemmatize() function to do this.

5. **Splitting the data into Train/Test sets:** The dataset is divided into training and testing sets using the train_test_split() function from sklearn. The testing set is used to assess the model's performance, while the training set is used to train the model.

Handling HTML tags, special characters, and various word forms (such as "run", "running", and "ran") and their variations are among the difficulties connected with this preprocessing. The chosen collection of preprocessing methods, however, is suitable for the specified purpose of sentiment analysis on the IMDB dataset. The methods employed are popular in NLP and have been demonstrated to increase the models' accuracy in a variety of tasks.

7 Baselines

The two baselines that were chosen for this task were the majority label and the random baseline. According to the majority label baseline, all test data points will have the label that appears most frequently in the training data as their predicted label. This baseline can be used as a helpful benchmark to determine whether the model can outperform the most prevalent label. Moreover, for the random baseline model each test data point’s predicted label is chosen at random from one of the labels in the training set. Any model that performs worse than random is deemed to be performing poorly. This baseline gives a lower constraint on performance.

These baselines are helpful for the task and dataset since they offer an easy way to compare the performance of the suggested models to a straightforward strategy. If the suggested models fall short of these straightforward baselines, there may be a fault with the model or more sophisticated methodologies may be needed.

8 Results, error analysis

The models in the NLP pipeline perform as follows on the IMDB dataset:

1. Baseline majority label model: The majority label is 0, and predicting all samples as 0 has an accuracy of 0.4961.

2. Random baseline model: When predicting labels at random, accuracy is 0.4966.

3. Bag of Words logistic regression model: the model obtains an accuracy of 0.87**.** For class 0 and class 1, precision, recall, and F1-score of 0.90, 0.83, 0.86, and 0.84, 0.91, 0.88 respectively.

4. The TF-IDF logistic regression model: the model obtains an accuracy of 0.89**.** For class 0 and class 1, precision, recall, and F1-score of 0.90, 0.88, 0.89, and 0.88, 0.91, 0.89 respectively.

5. Bag of Words naïve-bayes model: The model achieves an accuracy of 0.86. For class 0 and class 1, precision, recall, and F1-score of 0.82, 0.91, 0.86, and 0.90, 0.81, 0.85 respectively.

6. The TF-IDF naïve-bayes model: The model obtains an accuracy of 0.86. For class 0 and class 1, precision, recall, and F1-score of 0.85, 0.88, 0.86, and 0.87, 0.85, 0.86 respectively.

7. Bag of Words model (SVM): The model achieves an accuracy of 0.88. For class 0 and class 1, precision, recall, and F1-score of 0.92, 0.84, 0.88, and 0.85, 0.92, 0.89 respectively.

8. The TF-IDF model (SVM): The model obtains an accuracy of 0.89. For class 0 and class 1, precision, recall, and F1-score of 0.90, 0.88, 0.89, and 0.88, 0.90, 0.89 respectively.

All of the pipeline models perform significantly better than the baseline models. For a binary classification problem with balanced classes, an accuracy of about 0.5 is attained by the majority label baseline and random baseline models. The accuracy of every model in the pipeline is higher than 0.86, doing much better than the baseline.

The project's findings demonstrate how effectively the NLP models on the IMDB dataset performed. The TF-IDF logistic regression model had the best results, achieving an accuracy of 0.89. This is a positive outcome and indicates that the algorithm can successfully categorize movie reviews as positive or negative based on the text content.

Overall, the findings advance the project's objectives by shedding light on the strengths and weaknesses of the NLP models, which may be used to enhance them in the future and apply them to real-world problems.

Lessons learned and conclusions

I have gained useful knowledge and learned crucial lessons about setting up an NLP pipeline for sentiment analysis on the IMDB dataset throughout this project. In this situation, I found the TF-IDF and Bag of Words techniques to be quite helpful. The logistic regression and SVM classifiers also did well, reaching accuracy rates of 87% and 89%, respectively. The fact that these outcomes greatly outperformed the majority label and random baselines demonstrates the potency of the machine learning models.

I did, however, run into some unanticipated difficulties, such as coping with the noisy and ambiguous language and the trade-off between model complexity and generalizability. To overcome this, various feature engineering techniques, bag of words and TF-IDF, and hyperparameter tuning were experimented with.

Regarding the project's achievements, I think I was successful in achieving my initial aims and targets, which were to create a sentiment analysis pipeline for the IMDB dataset and assess the effectiveness of several machine learning models. I recognize that there is still potential for improvement, and to further enhance the outcomes, I could have investigated more sophisticated techniques like deep learning models or ensemble methods.

References

- K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, 2014.

- D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations (ICLR), 2015.

- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), pages 5998–6008, 2017.

- P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146, 2017.

- J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 1(8):9, 2019.

- Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

- D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

- M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.